ACL 2007

Proceedings of the Interactive Poster and Demonstration Sessions

June 25–27, 2007

Prague, Czech Republic

Production and Manufacturing by
Omnipress
2600 Anderson Street
Madison, WI 53704
USA

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
209 N. Eighth Street
Stroudsburg, PA 18360
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

Preface

The 45th Annual Meeting of the Association for Computational Linguistics, Posters and Demonstrations session was held between the 25th to 27th June 2007 in Prague. This year we had 113 submissions out of which 61 were selected for presentation, resulting in a 54% acceptance rate.

The criteria for acceptance of posters were to describe original work in progress, and to present innovative methodologies used to solve problems in computational linguistics or NLP. 48 posters were accepted.

For demonstrations the criterion for acceptance was the implementation of mature systems or prototypes in which computational linguistics or NLP technologies are used to solve practically important problems. 13 demonstrations were accepted.

I would like to thank the General Conference Chair of ACL 2007, John Carroll, for his insightful suggestions in formulating the call for papers. My gratitude to the members of the Program Committee for their promptness, professionalism and willingness in reviewing more papers than anticipated.

I would like to extend my thanks to the local organisers who accommodated a number of requests speedily making sure that the scheduling and the physical facilities were in place for this event. Last but not least, my special thanks to Scott Piao and Yutaka Sasaki for their help in the preparation of the camera-ready copy of the proceedings.

Sophia Ananiadou
Chair

Organizers

Chair:

Sophia Ananiadou, University of Manchester (UK)

Program Committee:

Timothy Baldwin, University of Melbourne (Australia)
Srinivas Bangalore, AT&, (USA)
Roberto Basili, University of Rome Tor Vergata (Italy)
Walter Daelemans, University of Antwerp (Belgium)
Beatrice Daille, Universite de Nantes (France)
Tomaz Erjavec, Jozef Stefan Institute in Ljubljana (Slovenia)
Katerina Frantzi, University of Aegean (Greece)
Sanda Harabagiu, University of Texas at Dallas (USA)
Jerry Hobbs, USC/ISI (USA)
Alessandro Lenci, Universita di Pisa (Italy)
Evangelos Milios, Dalhousie University (Canada)
Yusuke Miyao, University of Tokyo (Japan)
Kemal Oflazer, Sabanci University (Turkey)
Stelios Piperidis, ILSP (Greece)
Thierry Poibeau, Universite Paris 13 (France)
Paul Rayson, University of Lancaster (UK)
Philip Resnik, University of Maryland (USA)
Fabio Rinaldi, University of Zurich (Switzerland)
Anne de Roeck, Open University (UK)
Frederique Segond, Xerox Research Centre Europe (France)
Kumiko Tanaka-Ishii, University of Tokyo (Japan)
Kentaro Torisawa, JAIST (Japan)
Yoshimasa Tsuruoka, University of Manchester (UK)
Lucy Vanderwende, Microsoft (USA)
Pierre Zweigenbaum, Universite Paris XI (France)

Table of Contents

Program

Demos

Tuesday, June 26

9:00-10:40 **Demo Session 1**

MIMUS: A Multimodal and Multilingual Dialogue System for the Home Domain
J. Gabriel Amores, Guillermo Pérez and Pilar Manchón

A Translation Aid System with a Stratified Lookup Interface
Takeshi Abekawa and Kyo Kageura

Multimedia Blog Creation System using Dialogue with Intelligent Robot
Akitoshi Okumura, Takahiro Ikeda, Toshihiro Nishizawa, Shin-ichi Ando and Fumihiro Adachi

SemTAG: a platform for specifying Tree Adjoining Grammars and performing TAG-based Semantic Construction
Claire Gardent and Yannick Parmentier

System Demonstration of On-Demand Information Extraction
Satoshi Sekine and Akira Oda

Multilingual Ontological Analysis of European Directives
Gianmaria Ajani, Guido Boella, Leonardo Lesmo, Alessandro Mazzei and Piercarlo Rossi

13:30-15:20 **Demo Session 2**

zipfR: Word Frequency Modeling in R
Stefan Evert and Marco Baroni

Linguistically Motivated Large-Scale NLP with C&C and Boxer
James Curran, Stephen Clark and Johan Bos

Don't worry about metaphor: affect detection for conversational agents
Catherine Smith, Timothy Rumbell, John Barnden, Robert Hendley, Mark Lee, Alan Wallington and Li Zhang

An efficient algorithm for building a distributional thesaurus (and other Sketch Engine developments)
Pavel Rychlý and Adam Kilgarriff

Tuesday, June 26

Semantic enrichment of journal articles using chemical named entity recognition
Colin R. Batchelor and Peter T. Corbett

An API for Measuring the Relatedness of Words in Wikipedia
Simone Paolo Ponzetto and Michael Strube

Posters

Monday, June 25

15:10-15:45 **Poster Session 1**

Machine Learning, Corpus and Information Retrieval

Deriving an Ambiguous Words Part-of-Speech Distribution from Unannotated Text
Reinhard Rapp

Support Vector Machines for Query-focused Summarization trained and evaluated on Pyramid data
Maria Fuentes, Enrique Alfonseca and Horacio Rodríguez

A Joint Statistical Model for Simultaneous Word Spacing and Spelling Error Correction for Korean
Hyungjong Noh, Jeong-Won Cha and Gary Geunbae Lee

An Approximate Approach for Training Polynomial Kernel SVMs in Linear Time
Yu-Chieh Wu, Jie-Chi Yang and Yue-Shi Lee

Rethinking Chinese Word Segmentation: Tokenization, Character Classification, or Word-break Identification
Chu-Ren Huang, Petr Šimon, Shu-Kai Hsieh and Laurent Prévot

A Feature Based Approach to Leveraging Context for Classifying Newsgroup Style Discussion Segments
Yi-Chia Wang, Mahesh Joshi and Carolyn Rose

Ensemble document clustering using weighted hypergraph generated by NMF
Hiroyuki Shinnou and Minoru Sasaki

Using Error-Correcting Output Codes with Model-Refinement to Boost Centroid Text Classifier
Songbo Tan

Monday, June 25

Poliqarp: An open source corpus indexer and search engine with syntactic extensions
Daniel Janus and Adam Przepiórkowski

Test Collection Selection and Gold Standard Generation for a Multiply-Annotated Opinion Corpus
Lun-Wei Ku, Yong-Sheng Lo and Hsin-Hsi Chen

Generating Usable Formats for Metadata and Annotations in a Large Meeting Corpus
Andrei Popescu-Belis and Paula Estrella

Exploration of Term Dependence in Sentence Retrieval
Keke Cai, Jiajun Bu, Chun Chen and Kangmiao Liu

Minimum Bayes Risk Decoding for BLEU
Nicola Ehling, Richard Zens and Hermann Ney

10:40-11:10 Poster Session 2

Speech Dialogue

Disambiguating Between Generic and Referential "You" in Dialog
Surabhi Gupta, Matthew Purver and Dan Jurafsky

On the formalization of Invariant Mappings for Metaphor Interpretation
Rodrigo Agerri, John Barnden, Mark Lee and Alan Wallington

Real-Time Correction of Closed-Captions
Patrick Cardinal, Gilles Boulianne, Michel Comeau and Maryse Boisvert

Learning to Rank Definitions to Generate Quizzes for Interactive Information Presentation
Ryuichiro Higashinaka, Kohji Dohsaka and Hideki Isozaki

Predicting Evidence of Understanding by Monitoring User's Task Manipulation in Multimodal Conversations
Yukiko Nakano, Kazuyoshi Murata, Mika Enomoto, Yoshiko Arimoto, Yasuhiro Asa and Hirohiko Sagawa

Automatically Assessing the Post Quality in Online Discussions on Software
Markus Weimer, Iryna Gurevych and Max Mühlhäuser

Monday, June 25

NICT-ATR Speech-to-Speech Translation System
Eiichiro Sumita, Tohru Shimizu and Satoshi Nakamura

WordNet-based Semantic Relatedness Measures in Automatic Speech Recognition for Meetings
Michael Pucher

Tuesday, June 26

15:20-15:45 **Poster Session 3**

Lexica and Ontologies

Building Emotion Lexicon from Weblog Corpora
Changhua Yang, Kevin Hsin-Yih Lin and Hsin-Hsi Chen

Construction of Domain Dictionary for Fundamental Vocabulary
Chikara Hashimoto and Sadao Kurohashi

Extracting Word Sets with Non-Taxonomical Relation
Eiko Yamamoto and Hitoshi Isahara

A Linguistic Service Ontology for Language Infrastructures
Yoshihiko Hayashi

Empirical Measurements of Lexical Similarity in Noun Phrase Conjuncts
Deirdre Hogan

Automatic Discovery of Named Entity Variants: Grammar-driven Approaches to Non-Alphabetical Transliterations
Chu-Ren Huang, Petr Šimon and Shu-Kai Hsieh

Detecting Semantic Relations between Named Entities in Text Using Contextual Features
Toru Hirano, Yoshihiro Matsuo and Genichiro Kikui

Mapping Concrete Entities from PAROLE-SIMPLE-CLIPS to ItalWordNet: Methodology and Results
Adriana Roventini, Nilda Ruimy, Rita Marinelli, Marisa Ulivieri and Michele Mammini

Extracting Hypernym Pairs from the Web
Erik Tjong Kim Sang

Tuesday, June 26

An OWL Ontology for HPSG
Graham Wilcock

Wednesday, June 27

10:40-11:10 Poster Session 4

Applications

Classifying Temporal Relations Between Events
Nathanael Chambers, Shan Wang and Dan Jurafsky

Moses: Open Source Toolkit for Statistical Machine Translation
Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin and Evan Herbst

Boosting Statistical Machine Translation by Lemmatization and Linear Interpolation
Ruiqiang Zhang and Eiichiro Sumita

Extractive Summarization Based on Event Term Clustering
Maofu Liu, Wenjie Li, Mingli Wu and Qin Lu

Machine Translation between Turkic Languages
Ahmet Cüneyd Tantuğ, Esref Adali and Kemal Oflazer

Measuring Importance and Query Relevance in Topic-focused Multi-document Summarization
Surabhi Gupta, Ani Nenkova and Dan Jurafsky

Expanding Indonesian-Japanese Small Translation Dictionary Using a Pivot Language
Masatoshi Tsuchiya, Ayu Purwarianti, Toshiyuki Wakita and Seiichi Nakagawa

Wednesday, June 27

15:10-15:45 **Poster Session 5**

Parsing and Tagging

Shallow Dependency Labeling
Manfred Klenner

Minimally Lexicalized Dependency Parsing
Daisuke Kawahara and Kiyotaka Uchimoto

HunPos – an open source trigram tagger
Péter Halácsy, András Kornai and Csaba Oravecz

Extending MARIE: an N-gram-based SMT decoder
Josep M. Crego and José B. Mariño

A Hybrid Approach to Word Segmentation and POS Tagging
Tetsuji Nakagawa and Kiyotaka Uchimoto

Automatic Part-of-Speech Tagging for Bengali: An Approach for Morphologically Rich Languages in a Poor Resource Scenario
Sandipan Dandapat, Sudeshna Sarkar and Anupam Basu

Japanese Dependency Parsing Using Sequential Labeling for Semi-spoken Language
Kenji Imamura, Genichiro Kikui and Norihito Yasuda

MIMUS: A Multimodal and Multilingual Dialogue System for the Home Domain

J. Gabriel Amores
Julietta Research Group
Universidad de Sevilla
jgabriel@us.es

Guillermo Pérez
Julietta Research Group
Universidad de Sevilla
gperez@us.es

Pilar Manchón
Julietta Research Group
Universidad de Sevilla
pmanchon@us.es

Abstract

This paper describes MIMUS, a multimodal and multilingual dialogue system for the in–home scenario, which allows users to control some home devices by voice and/or clicks. Its design relies on Wizard of Oz experiments and is targeted at disabled users. MIMUS follows the Information State Update approach to dialogue management, and supports English, German and Spanish, with the possibility of changing language on–the–fly. MIMUS includes a gestures–enabled talking head which endows the system with a human–like personality.

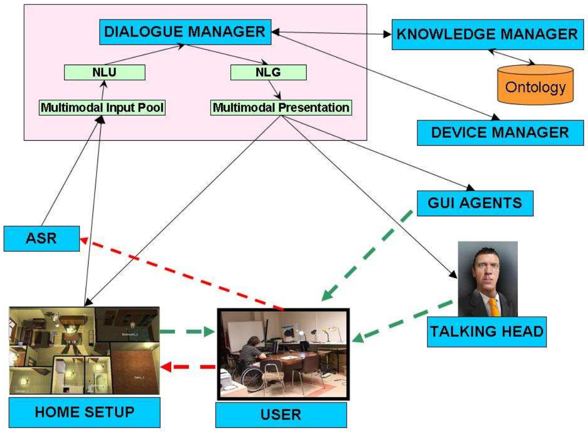


Figure 1: MIMUS Architecture

1 Introduction

This paper describes MIMUS, a multimodal and multilingual dialogue system for the in–home scenario, which allows users to control some home devices by voice and/or clicks. The architecture of MIMUS was first described in (Pérez et al., 2006c). This work updates the description and includes a life demo. MIMUS follows the Information State Update approach to dialogue management, and has been developed under the EU–funded TALK project (Talk Project, 2004). Its architecture consists of a set of OAA agents (Cheyer and Martin, 1972) linked through a central Facilitator, as shown in figure 1:

The main agents in MIMUS are briefly described hereafter:

- The system core is the **Dialogue Manager**, which processes the information coming from the different input modality agents by means of a natural language understanding module and provides output in the appropriate modality.

- The main input modality agent is the **ASR Manager**, which is obtained through an OAA wrapper for Nuance. Currently, the system supports English, Spanish and German, with the possibility of changing languages on–the–fly without affecting the dialogue history.

- The **HomeSetup** agent displays the house layout, with all the devices and their state. Whenever a device changes its state, the HomeSetup is notified and the graphical layout is updated.

- The **Device Manager** controls the physical devices. When a command is sent, the Device Manager notifies it to the HomeSetup and the Knowledge Manager, guaranteeing coherence in all the elements in MIMUS.

- The **GUI Agents** control each of the device–specific GUIs. Thus, clicking on the telephone icon, a telephone GUI will be displayed, and so on for each type of service.

- The **Knowledge Manager** connects all the agents to the common knowledge resource by

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 1–4,
Prague, June 2007. ©2007 Association for Computational Linguistics

means of an OWL Ontology.

- The **Talking Head**. MIMUS virtual character is synchronized with Loquendo's TTS, and has the ability to express emotions and play some animations such as nodding or shaking the head.

2 WoZ Experiments

MIMUS has been developed taking into account wheel–chair bound users. In order to collect first–hand information about the users' natural behavior in this scenario, several WoZ experiments were first conducted. A rather sophisticated multilingual WoZ experimental platform was built for this purpose.

The set of WoZ experiments conducted was designed in order to collect data. In turn, these data helped determine the relevant factors to configure multimodal dialogue systems in general, and MIMUS in particular.

A detailed description of the results obtained after the analysis of the experiments and their impact on the overall design of the system may be found in (Manchón et al., 2007).

3 ISU–based Dialogue Management in MIMUS

As pointed out above, MIMUS follows the ISU approach to dialogue management (Larsson and Traum, 2000). The main element of the ISU approach in MIMUS is the dialogue history, represented formally as a list of dialogue states. Dialogue rules update this information structure either by producing new dialogue states or by supplying arguments to existing ones.

3.1 Multimodal DTAC structure

The information state in MIMUS is represented as a feature structure with four main attributes: **D**ialogue Move, **T**ype, **A**rguments and **C**ontents.

- **DMOVE**: Identifies the kind of dialogue move.
- **TYPE**: This feature identifies the specific dialogue move in the particular domain at hand.
- **ARGS**: The `ARGS` feature specifies the argument structure of the `DMOVE/TYPE` pair.

Modality and Time features have been added in order to implement fusion strategies at dialogue level.

3.2 Updating the Information State in MIMUS

This section provides an example of how the Information State Update approach is implemented in MIMUS. Update rules are triggered by dialogue moves (any dialogue move whose DTAC structure unifies with the Attribute–Value pairs defined in the TriggeringCondition field) and may require additional information, defined as dialogue expectations (again, those dialogue moves whose DTAC structure unify with the Attribute–Value pairs defined in the DeclareExpectations field).

Consider the following DTAC, which represents the information state returned by the NLU module for the sentence *switch on*:

$$\begin{bmatrix} \text{DMOVE} & \text{specifyCommand} \\ \text{TYPE} & \text{SwitchOn} \\ \text{ARGS} & \begin{bmatrix}\text{Location, DeviceType}\end{bmatrix} \\ \text{META\_INFO} & \begin{bmatrix} \text{MODALITY} & \text{VOICE} \\ \text{TIME\_INIT} & 00{:}00{:}00 \\ \text{TIME\_END} & 00{:}00{:}30 \\ \text{CONFIDENCE} & 700 \end{bmatrix} \end{bmatrix}$$

Consider now the (simplified) dialogue rule "**ON**", defined as follows:

```
RuleID:    ON;
TriggeringCondition:
  (DMOVE:specifyCommand,
   TYPE:SwitchOn);
DeclareExpectations: {
  Location,
  DeviceType }
ActionsExpectations: {
  [DeviceType] =>
      {NLG(DeviceType);} }
PostActions: {
   ExecuteAction(@is-ON); }
```

The DTAC obtained for *switch on* triggers the dialogue rule **ON**. However, since two declared expectations are still missing (**Location** and **DeviceType**), the dialogue manager will activate the ActionExpectations and prompt the user for the kind of device she wants to switch on, by means of a call to the natural language generation module NLG(DeviceType). Once all expectations have been fulfilled, the PostActions can be executed over the desired device(s).

4 Integrating OWL in MIMUS

Initially, OWL Ontologies were integrated in MIMUS in order to improve its knowledge management module. This functionality implied the implementation of a new OAA wrapper capable of querying OWL ontologies, see (Pérez et al., 2006b) for details.

4.1 From Ontologies to Grammars: OWL2Gra

OWL ontologies play a central role in MIMUS. This role is limited, though, to the input side of the system. The domain–dependent part of multimodal and multilingual production rules for context–free grammars is semi–automatically generated from an OWL ontology.

This approach has achieved several goals: it leverages the manual work of the linguist, and ensures coherence and completeness between the Domain Knowledge (Knowledge Manager Module) and the Linguistic Knowledge (Natural Language Understanding Module) in the application. A detailed explanation of the algorithm and the results obtained can be found in (Pérez et al., 2006a)

4.2 From OWL to the House Layout

MIMUS home layout does not consist of a pre–defined static structure only usable for demonstration purposes. Instead, it is dynamically loaded at execution time from the OWL ontology where all the domain knowledge is stored, assuring the coherence of the layout with the rest of the system.

This is achieved by means of an OWL–RDQL wrapper. It is through this agent that the Home Setup enquires for the location of the walls, the label of the rooms, the location and type of devices per room and so forth, building the 3D graphical image from these data.

5 Multimodal Fusion Strategies

MIMUS approach to multimodal fusion involves combining inputs coming from different multimodal channels at dialogue level (Pérez et al., 2005). The idea is to check the multimodal input pool before launching the actions expectations while waiting for an "inter–modality" time. This strategy assumes that each individual input can be considered as an independent dialogue move. In this approach, the multimodal input pool receives and stores all inputs including information such as time and modality. The Dialogue Manager checks the input pool regularly to retrieve the corresponding input. If more than one input is received during a certain time frame, they are considered simultaneous or pseudo–simultaneous. In this case, further analysis is needed in order to determine whether those independent multimodal inputs are truly related or not. Another, improved strategy has been proposed at (Manchón et al., 2006), which combines the advantages of this one, and those proposed for unification–based grammars (Johnston et al., 1997; Johnston, 1998).

6 Multimodal Presentation in MIMUS

MIMUS offers graphical and voice output to the users through an elaborate architecture composed of a TTS Manager, a HomeSetup and GUI agents. The multimodal presentation architecture in MIMUS consists of three sequential modules. The current version is a simple implementation that may be extended to allow for more complex theoretical issues hereby proposed. The main three modules are:

- Content Planner (CP): This module decides on the information to be provided to the user. As pointed out by (Wahlster et al., 1993), the CP cannot determine the content independently from the presentation planner (PP). In MIMUS, the CP generates a set of possibilities, from which the PP will select one, depending on their feasibility.

- Presentation Planner (PP): The PP receives the set of possible content representations and selects the "best" one.

- Realization Module (RM): This module takes the presentation generated and selected by the CP–PP, divides the final DTAC structure and sends each substructure to the appropriate agent for rendering.

7 The MIMUS Talking Head

MIMUS virtual character is known as *Ambrosio*. Endowing the character with a name results in personalization, personification, and voice activation. Ambrosio will remain inactive until called for duty (voice activation); each user may name their personal assistant as they wish (Personalization); and they will address the system at personal level, reinforcing the sense of human–like communication (Personification). The virtual head has been implemented in 3D to allow for more natural and realistic gestures and movements. The graphical engine used is OGRE (OGRE, 2006), a powerful, free and easy to use tool. The current talking head is integrated with Loquendo, a high quality commercial synthesizer that launches the information about the phonemes as asynchronous events, which allows for lip synchronization. The dialogue manager controls the talking head, and sends the appropriate commands depending of the dialogue needs. Throughout the dialogue, the dialogue manager may see it fit to reinforce the communication channel with gestures and expressions, which may or may not imply synthesized utterances. For instance, the head may just nod to acknowledge a command, without uttering words.

8 Conclusions and Future Work

In this paper, an overall description of the MIMUS system has been provided.

MIMUS is a fully multimodal and multilingual dialogue system within the Information State Update approach. A number of theoretical and practical issues have been addressed successfully, resulting in a user–friendly, collaborative and humanized system.

We concluded from the experiments that a human–like talking head would have a significant positive impact on the subjects' perception and willingness to use the system.

Although no formal evaluation of the system has taken place, MIMUS has already been presented successfully in different forums, and as expected, "Ambrosio" has always made quite an impression, making the system more appealing to use and approachable.

References

Adam Cheyer and David Martin. 2001. The open agent architecture. *Journal of Autonomous Agents and Multi–Agent Systems*, 4(12):143–148.

Michael Johnston, Philip R. Cohen, David McGee, Sharon L. Oviatt, James A. Pitman and Ira A. Smith. 1997. Unification–based Multimodal Integration *ACL* 281–288.

Michael Johnston. 1998. Unification–based Multimodal Parsing *Coling–ACL* 624–630.

Staffan Larsson and David Traum. 2000. Information State and dialogue management in the TRINDI Dialogue Move Engine Toolkit. *Natural Language Engineering*, 6(34): 323-340.

Pilar Manchón, Guillermo Pérez and Gabriel Amores. 2006. Multimodal Fusion: A New Hybrid Strategy for Dialogue Systems. *Proceedings of International Congress of Multimodal Interfaces (ICMI06)*, 357–363. ACM, New York, USA.

Pilar Manchón, Carmen Del Solar, Gabriel Amores and Guillermo Pérez. 2007. Multimodal Event Analysis in the MIMUS Corpus. *Multimodal Corpora: Special Issue of the International Journal JLRE*, submitted.

OGRE. 2006. Open Source Graphics Engine. *www.ogre3d.org*

Guillermo Pérez, Gabriel Amores and Pilar Manchón. 2005. Two Strategies for multimodal fusion. E.V. Zudilova–Sainstra and T. Adriaansen (eds.) *Proceedings of Multimodal Interaction for the Visualization and Exploration of Scientific Data*, 26–32. Trento, Italy.

Guillermo Pérez, Gabriel Amores, Pilar Manchón and David González Maline. 2006. Generating Multilingual Grammars from OWL Ontologies. *Research in Computing Science*, 18:3–14.

Guillermo Pérez, Gabriel Amores, Pilar Manchón, Fernando Gómez and Jesús González. 2006. Integrating OWL Ontologies with a Dialogue Manager. *Procesamiento del Lenguaje Natural* 37:153–160.

Guillermo Pérez, Gabriel Amores and Pilar Manchón. 2006. A Multimodal Architecture For Home Control By Disabled Users. *Proceedings of the IEEE/ACL 2006 Workshop on Spoken Language Technology*, 134–137. IEEE, New York, USA.

Talk Project. Talk and Look: Linguistic Tools for Ambient Linguistic Knowledge. 2004. 6th Framework Programme. *www.talk-project.org*

Wolfgang Wahlster, Elisabeth André, Wolfgang Finkler, Hans–Jürgen Profitlich and Thomas Rist. 1993. Plan–Based integration of natural language and graphics generation. *Artificial intelligence*, 63:287–247.

A Translation Aid System with a Stratified Lookup Interface

Takeshi Abekawa and **Kyo Kageura**
Library and Information Science Course
Graduate School of Education,
University of Tokyo, Japan
{abekawa,kyo}@p.u-tokyo.ac.jp

Abstract

We are currently developing a translation aid system specially designed for English-to-Japanese volunteer translators working mainly online. In this paper we introduce the stratified reference lookup interface that has been incorporated into the source text area of the system, which distinguishes three user awareness levels depending on the type and nature of the reference unit. The different awareness levels are assigned to reference units from a variety of reference sources, according to the criteria of "composition", "difficulty", "speciality" and "resource type".

1 Introduction

A number of translation aid systems have been developed so far (Bowker, 2002; Gow, 2003). Some systems such as TRADOS have proved useful for some translators and translation companies[1]. However, volunteer (and in some case freelance) translators do not tend to use these systems (Fulford and Zafra, 2004; Fulford, 2001; Kageura et al., 2006), for a variety of reasons: most of them are too expensive for volunteer translators[2]; the available functions do not match the translators' needs and work style; volunteer translators are under no pressure from clients to use the system, etc. This does not mean, however, that volunteer translators are satisfied with their working environment.

Against this backdrop, we are developing a translation aid system specially designed for English-to-Japanese volunteer translators working mainly online. This paper introduces the stratified reference lookup/notification interface that has been incorporated into the source text area of the system, which distinguishes three user awareness levels depending on the type and nature of the reference unit. We show how awareness scores are given to the reference units and how these scores are reflected in the way the reference units are displayed.

2 Background

2.1 Characteristics of target translators

Volunteer translators involved in translating English online documents into Japanese have a variety of backgrounds. Some are professional translators, some are interested in the topic, some translate as a part of their NGO activities, etc[3]. They nevertheless share a few basic characteristics: (i) they are native speakers of Japanese (the target language: TL); (ii) most of them do not have a native-level command in English (the source language: SL); (iii) they do not use a translation aid system or MT; (iv) they want to reduce the burden involved in the process of translation; (v) they spend a huge amount of time looking up reference sources; (vi) the smallest basic unit of translation is the paragraph and "at a glance" readability of the SL text is very important. A translation aid system for these translators should provide enhanced and easy-to-use reference lookup functions with quality reference sources. An important point expressed by some translators is that they do not want a system that makes decisions on their behalf; they want the system to help them make decisions by making it easier for them to access references. Decision-making by translations in fact constitutes an essential part of the translation process (Munday, 2001; Venuti, 2004).

[1] http://www.trados.com/

[2] Omega-T, http://www.omegat.org/

[3] We carried out a questionnaire survey of 15 volunteer translators and interviewed 5 translators.

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 5–8,
Prague, June 2007. ©2007 Association for Computational Linguistics

Some of these characteristics contrast with those of professional translators, for instance, in Canada or in the EU. They have native command in both the source and target languages; they went through university-level training in translation; many of them have a speciality domain; they work on the principle that "time is money" [4]. For this type of translator, facilitating target text input can be important, as is shown in the TransType system (Foster et al., 2002; Macklovitch, 2006).

2.2 Reference units and lookup patterns

The major types of reference unit can be summarised as follows (Kageura et al., 2006).

Ordinary words: Translators are mostly satisfied with the information provided in existing dictionaries. Looking up these references is not a huge burden, though reducing it would be preferable.

Idioms and phrases: Translators are mostly satisfied with the information provided in dictionaries. However, the lookup process is onerous and many translators worry about failing to recognise idioms in SL texts (as they can often be interpreted literally), which may lead to mistranslations.

Technical terms: Translators are not satisfied with the available reference resources [5]; they tend to search the Internet directly. Translators tend to be concerned with failing to recognise technical terms.

Proper names: Translators are not satisfied with the available reference resources. They worry more about misidentifying the referent. For the identification of the referent, they rely on the Internet.

3 The translation aid system: QRedit

3.1 System overview

The system we are developing, QRedit, has been designed with the following policies: making it less onerous for translators to do what they are currently doing; providing information efficiently to facilitate decision-making by translators; providing functions in a manner that matches translators' behaviour.

QRedit operates on the client server model. It is implemented by Java and run on Tomcat. Users access the system through Web browsers. The integrated editor interface is divided into two main areas: the SL text area and the TL editing area. These scroll synchronically. To enable translators to maintain their work rhythm, the keyboard cursor is always bound to the TL editing area (Abekawa and Kageura, 2007).

3.2 Reference lookup functions

Reference lookup functions are activated when an SL text is loaded. Relevant information (translation candidates and related information) is displayed in response to the user's mouse action. In addition to simple dictionary lookup, the system also provides flexible multi-word unit lookup mechanisms. For instance, it can automatically look up the dictionary entry "with one's tongue in one's cheek" for the expression "He said that *with his* big fat *tongue in his* big fat *cheek*" or "head screwed on *right*" for "head screwed on *wrong*" (Kanehira et al., 2006).

The reference information can be displayed in two ways: a simplified display in a small popup window that shows only the translation candidates, and a full display in a large window that shows the full reference information. The former is for quick reference and the latter for in-depth examination.

Currently, *Sanseido's Grand Concise English-Japanese Dictionary*, *Eijiro*[6], List of technical terms in 23 domains, and Wikipedia are provided as reference sources.

4 Stratified reference lookup interface

In relation to reference lookup functions, the following points are of utmost importance:

1. In the process of translation, translators often check multiple reference resources and examine several meanings in SL and expressions in TL. We define the provision of "good information" for the translator by the system as information that the translator can use to make his or her own decisions.

2. The system should show the range of available information in a manner that corresponds to the translator's reference lookup needs and behaviour.

[4] Personal communication with Professor Elliott Macklovitch at the University of Montreal, Canada.

[5] With the advent of Wikipedia, this problem is gradually becoming less important.

[6] http://www.eijiro.jp/

The reference lookup functions can be divided into two kinds: (i) those that notify the user of the existence of the reference unit, and (ii) those that provide reference information. Even if a linguistic unit is registered in reference sources, if the translator is unaware of its existence, (s)he will not look up the reference, which may result in mistranslation. It is therefore preferable for the system to notify the user of the possible reference units. On the other hand, the richer the reference sources become, the greater the number of candidates for notification, which would reduce the readability of SL texts dramatically. It was necessary to resolve this conflict by striking an appropriate balance between the notification function and user needs in both reference lookup and the readability of the SL text.

4.1 Awareness levels

To resolve this conflict, we introduced three translator "awareness levels":

- Awareness level -2: Linguistic units that the translator may not notice, which will lead to mistranslation. The system always actively notifies translators of the existence of this type of unit, by underlining it. Idioms and complex technical terms are natural candidates for this awareness level.
- Awareness level -1: Linguistic units that translators may be vaguely aware of or may suspect exist and would like to check. To enable the user to check their existence easily, the relevant units are displayed in bold when the user moves the cursor over the relevant unit or its constituent parts with the mouse. Compounds, easy idioms and fixed expressions are candidates for this level.
- Awareness level 0: Linguistic units that the user can always identify. Single words and easy compounds are candidates for this level.

In all these cases, the system displays reference information when the user clicks on the relevant unit with the mouse.

4.2 Assignment of awareness levels

The awareness levels defined above are assigned to the reference units on the basis of the following four characteristics:

***C(unit)*:** The compositional nature of the unit. Single words can always be identified in texts, so the score 0 is assigned to them. The score -1 is assigned to compound units. The score -2 is assigned to idioms and compound units with gaps.

***D(unit)*:** The difficulty of the linguistic unit for a standard volunteer translator. For units in the list of elementary expressions[7], the score 1 is given. The score 0 is assigned to words, phrases and idioms listed in general dictionaries. The score -1 is assigned to units registered only in technical term lists.

***S(unit)*:** The degree of domain dependency of the unit. The score -1 is assigned to units that belong to the domain which is specified by the user. The score 0 is assigned to all the other units. The domain information is extracted from the domain tags in ordinary dictionaries and technical term lists. For Wikipedia entries the category information is used.

***R(unit)*:** The type of reference source to which the unit belongs. We distinguish between dictionaries and encyclopaedia, corresponding to the user's information search behaviour. The score -1 is assigned to units which are registered in the encyclopaedia (currently Wikipedia[8]), because the fact that factual information is registered in existing reference sources implies that there is additional information relating to these units which the translator might benefit from knowing. The score 0 is assigned to units in dictionaries and technical term lists.

The overall score $A(unit)$ for the awareness level of a linguistic unit is calculated by:

$$A(unit) = C(unit) + D(unit) + S(unit) + R(unit).$$

Table 1 shows the summary of awareness levels and the scores of each characteristic. For instance, in an the SL sentence "The airplane *took* right *off*.", the $C(\text{take off}) = -2$, $D(\text{take off}) = 1$, $S(\text{take off}) = 0$ and $R(\text{take off}) = 0$; hence $A(\text{take off}) = -1$.

A score lower than -2 is normalised to -2, and a score higher than 0 is normalised to 0, because we assume three awareness levels are convenient for realising the corresponding notification interface and

[7] This list consists of 1,654 idioms and phrases taken from multiple sources for junior high school and high school level English reference sources published in Japan.

[8] As the English Wikipedia has entries for a majority of ordinary words, we only assign the score -1 to proper names.

$A(unit)$: awareness level	<= -2	-1	>= 0	
Mode of alert	always emphasis	by mouse-over	none	
Score	-2	-1	0	1
$C(unit)$: composition	compound unit with gap	compound unit	single word	
$D(unit)$: difficulty		technical term	general term	elementary term
$S(unit)$: speciality		specified domain	general domain	
$R(unit)$: resource type		encyclopaedia	dictionary	

Table 1: Awareness levels and the scores of each characteristic

are optimal from the point of view of the user's search behaviour. We are currently examining user customisation functions.

5 Conclusion

In this paper, we introduced a stratified reference lookup interface within a translation aid environment specially designed for English-to-Japanese online volunteer translators. We described the incorporation into the system of different "awareness levels" for linguistic units registered in multiple reference sources in order to optimise the reference lookup interface. The incorporation of these levels stemmed from the basic understanding we arrived at after consulting with actual translators that functions should fit translators' actual behaviour. Although the effectiveness of this interface is yet to be fully examined in real-world situations, the basic concept should be useful as the idea of awareness level comes from feedback by monitors who used the first version of the system.

Although in this paper we focused on the use of established reference resources, we are currently developing (i) a mechanism for recycling relevant existing documents, (ii) dynamic lookup of proper name transliteration on the Internet, and (iii) dynamic detection of translation candidates for complex technical terms. How to fully integrate these functions into the system is our next challenge.

References

Takeshi Abekawa and Kyo Kageura. 2007. Qredit: An integrated editor system to support online volunteer translators. In *Proceedings of Digital Humanities 2007 Poster/Demos*.

Lynne Bowker. 2002. *Computer-aided Translation Technology: A Practical Introduction*. Ottawa: University of Ottawa Press.

George Foster, Philippe Langlais, and Guy Lapalme. 2002. User-friendly text prediction for translators. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 148–155.

Heather Fulford and Joaquín Granell Zafra. 2004. The uptake of online tools and web-based language resources by freelance translators. In *Proceedings of the Second International Workshop on Language Resources for Translation Work, Research and Training*, pages 37–44.

Heather Fulford. 2001. Translation tools: An exploratory study of their adoption by UK freelance translators. *Machine Translation*, 16(3):219–232.

Francie Gow. 2003. *Metrics for Evaluating Translation Memory Software*. PhD thesis, Ottawa: University of Ottawa.

Kyo Kageura, Satoshi Sato, Koichi Takeuchi, Takehito Utsuro, Keita Tsuji, and Teruo Koyama. 2006. Improving the usability of language reference tools for translators. In *Proceedings of the 10th of Annual Meeting of Japanese Natural Language Processing*, pages 707–710.

Kou Kanehira, Kazuki Hirao, Koichi Takeuchi, and Kyo Kageura. 2006. Development of a flexible idiom lookup system with variation rules. In *Proceedings of the 10th Annual Meeting of Japanese Natural Language Processing*, pages 711–714.

Elliott Macklovitch. 2006. Transtype2: the last word. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC2006)*, pages 167–172.

Jeremy Munday. 2001. *Introducing Translation Studies: Theories and Applications*. London: Routledge.

Lawrence Venuti. 2004. *The Translation Studies Reader*. London: Routledge, second edition.

Multimedia Blog Creation System using Dialogue with Intelligent Robot

Akitoshi Okumura, Takahiro Ikeda, Toshihiro Nishizawa, Shin-ichi Ando, and Fumihiro Adachi
Common Platform Software Research Laboratries,
NEC Corporation
1753 Shimonumabe Nakahara-ku, Kawasaki-city, Kanagawa 211-8666 JAPAN
{a-okumura@bx,nishizawa@bk,s-ando@cw,f-adachi@aj}.jp.nec.com

Abstract

A multimedia blog creation system is described that uses Japanese dialogue with an intelligent robot. Although multimedia blogs are increasing in popularity, creating blogs is not easy for users who lack high-level information literacy skills. Even skilled users have to waste time creating and assigning text descriptions to their blogs and searching related multimedia such as images, music, and illustrations. To enable effortless and enjoyable creation of multimedia blogs, we developed the system on a prototype robot called PaPeRo. Video messages are recorded and converted into text descriptions by PaPeRo using continuous speech recognition. PaPeRo then searches for suitable multimedia contents on the internet and databases, and then, based on the search results, chooses appropriate sympathetic comments by using natural language text retrieval. The retrieved contents, PaPeRo's comments, and the video recording on the user's blog is automatically uploaded and edited. The system was evaluated by 10 users for creating travel blogs and proved to be helpful for both inexperienced and experienced users. The system enabled easy multimedia-rich blog creation and even provided users the pleasure of chatting with PaPeRo.

1 Introduction

Blogs have become popular and are used in a variety of settings not only for personal use, but are also used in the internal communications of organizations. A multimedia blog, which contains videos, music, and illustrations, is increasing in popularity because it enables users to express their thoughts creatively. However, users are unsatisfied with the current multimedia blog creation methods. Users have three requirements. First, they need easier methods to create blogs. Most multimedia blogs are created in one of two ways: 1) A user creates audio-visual contents by cameras and or some other recording devices, and then assigns a text description to the contents as indexes. 2) A user creates a text blog, and then searches for multimedia contents on the internet and databases to attach them to his blog. Both methods require high-level information literacy skills. Second, they would like to reduce their blog-creation time. Even skilled users have to waste time assigning text description and searching related multimedia contents. Third, they like to be encouraged by other peoples' comments on their blogs. Although some users utilize pet-type agents making automatic comments to their blogs, the agents do not always satisfy them because the comments do not consider users' moods. To meet the three requirements, we developed a multimedia blog creation system using Japanese dialogue with an intelligent robot. The system was developed on a prototype robot called PaPeRo (Fujita, 2002), which has the same CPU and memory as a mobile PC. In this paper, we describe the multimedia blog creation method and the evaluation results in a practical setting.

2 Multimedia Blog Creation

2.1 Outline of system processes

The system has four sequential processes: video message recording, continuous speech recognition, natural language text retrieval, and blog coordina-

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 9–12,
Prague, June 2007.

tion. The first process is activated when a user begins a conversation with PaPeRo. The process stores a video message recorded on PaPeRo's microphones and CCD cameras, and the second process converts the speech contents of the video message into a text description to extract important keywords. Then, the third process searches for suitable multimedia contents on pre-specified web sites and databases based on the text description. The first three processes can simplify multimedia blog creation and reduce creation costs. The last process detects a user's mood, such as delight, anger, sorrow, and pleasure, by extracting typical expressions from the text description, and then chooses appropriate sympathetic comments to encourage the user. Finally, the last process coordinates uploading the recorded video message, the text description, the extracted keywords, the searched contents, and the sympathetic comments on the user's blog.

2.2 Continuous Speech Recognition

The system converts the speech content of the video message into text descriptions and extracts important keywords based on their lexical information. The system should, therefore, be equipped with a large-vocabulary continuous speech recognition engine capable of dealing with spontaneous speech. This is because blog messages usually contain various kinds of words and expressions. As this kind of engine needs a large amount of memory and computational resources, it is generally difficult to install the engine on small intelligent robots because the robot requires its own computational resources for their own intelligent operations, such as image recognition and movement control. To solve this problem, we used a compact and scalable large-vocabulary continuous speech recognition framework, which has been shown to work on low-power devices, such as PDAs (Isotani et al.,2005). The framework achieves compact and high-speed processing by the following techniques:

- Efficient reduction of Gaussian components using MDL criterion (Shinoda, et al., 2002)
- High-speed likelihood calculation using tree-structured probability density functions (Watanabe, et al., 1995)
- Compaction of search algorithm using lexical prefix tree and shared usage of calculated language model scores (Isotani et al., 2005)

The framework we developed contained a Japanese lexicon of 50,000 words typically used in travel conversations based on a speech translation system (Isotani, et al., 2002). We were able to evaluate the developed system by making a travel blog using Japanese dialogue with PaPeRo.

2.3 Natural Language Text Retrieval

The system generates a query sentence from a text description converted using the above-mentioned framework. As few multimedia contents contain retrieval keywords, the system matches the query to text in web pages and documents containing multimedia contents. The system then chooses multimedia contents located adjacent to the highly-matched text as retrieval results. To achieve high precision for avoiding user interaction with the retrieved results, the system is enhanced using the Okapi BM25 model (Robertson, et al., 1995) by the following techniques (Ikeda, et al., 2005):

(1) Utilization of syntactic relationships

The system needs to distinguish illustrations based on the context. For example, an illustration of fish to be eaten in a restaurant should be different from that of fish to be seen in an aquarium. To achieve this, the system utilizes the syntactic relationships between a pair of words. The system produces a higher score for text containing the same syntactic relationship as that of a pair of words in a query sentence when calculating the matching score.

(2) Distinction of negation and affirmation

The system needs to distinguish negative and affirmative expressions because their meanings are clearly opposite. To achieve this, the system checks adjuncts attached to the expressions when matching a query sentence and text.

(3) Identification of synonyms

As different expressions have the same meaning, the system normalizes expressions by using a synonym dictionary containing 500 words before matching a query sentence and text.

2.4 Blog Coordination

The system detects users' mood to choose encouraging comments. Users' moods are sometimes detected by the expressions used and the manner in which the utterances are spoken. Although speaking manner can clearly detect emotions, such as laughing or crying, some emotions are not always indicated. Expressions that clearly identify a person's mood can be indicated (Nakamura, 1993). By studying moods that are easily detectable from expressions, including modality, we developed a database of 10 moods (delight, anger, sorrow, pleasure, desire, fear, shame, relief, surprise, and normal) individually linked with 100 kinds of specific expressions. The database is searched based on the above-mentioned natural language text retrieval, which considers syntactic relationships, negative and affirmative responses, and synonyms. The database is also linked to PaPeRo's response to convey the most appropriate sympathy for each mood. The response includes verbal comments, such as "I'm happy for you" and "It's really too bad", and gestures, such as dancing and crying depicted using GIF animation files. Responses are chosen based on the mood detected. Finally, the system coordinates uploading a recorded video message, the text description, the extracted important keywords, the searched multimedia contents, and PaPeRo's responses on the user's blog.

3 Example Use in Practical Setting

We developed a prototype system for creating a travel blog on PaPeRo, which can retrieve 2000 web pages containing 1500 illustrations and 550 songs. PaPeRo is activated by hearing the phrase, "can you help me make my blog please?", as listed in Table. 1, and creates a blog, as shown in Figure 1. Figure 1 shows a screen shot of a video message attached to the blog, a text description converted by the speech recognition and a button for playing the video message (A). Keywords, in this case Yosemite, Las Vegas, and Roulette, extracted from the text description are displayed (B). Three illustrations searched based on a query using the text description are displayed (C). A button for playing a searched song is available (D). PaPeRo's comments, such as "I hope that happens", are displayed (E). The user's mood is detected as desire from her saying "I ***would like to*** go there again." The comment is displayed together with the appropriate PaPeRo's response.

Table 1. Dialogue example (Excerpt)

A user : Can you help me make my blog please?
PaPeRo: Yes, please push the button on my head.
A user : I went to Yosemite for my winter vacation. I played the roulette for the first time in Las Vegas. I would like to go there again.
PaPeRo: Ok, now your blog is ready for viewing.

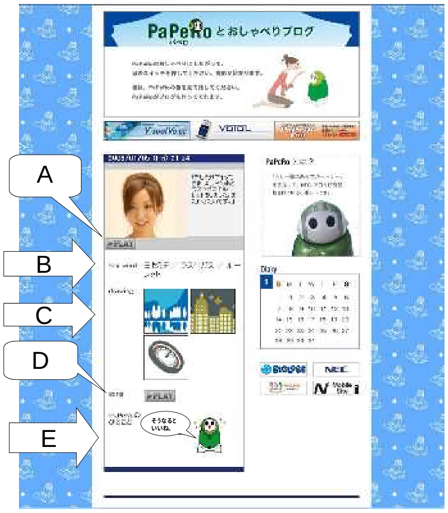


Figure 1. Example of Created Blog

4 Evaluation and Discussion

4.1 Evaluation

The system needs to be evaluated from two perspectives. The first is to individually evaluate the performance of each process mentioned in section 2. The second is to evaluate total performance, including the users' subjective opinions. As performance has been evaluated using different application systems, such as an automatic speech translation system (Isotani, et al., 2002) and a speech-activated text retrieval system (Ikeda, et al., 2005), we concentrated on evaluating the total performance based on surveying users' opinions about the blogs they created using the developed system. The survey results were analyzed in terms of speech recognition accuracy and users' blog making experience to improve the system.

4.2 Results and Discussion

The system was evaluated by 10 users. Half had blog making experiences, and the other half had no experience at all. All users input 20 sentences, and half of the sentences input were on travel issues, but the other half were unrelated because we needed opinions based on the results from low speech recognition accuracy. Users were interviewed on their automatically created blogs. Their opinions are listed in Table 2. The first row contains opinions about blogs created based on speech recognition results that had high word accuracy (85-95%). The second row contains opinions that had low accuracy (50-84%). The third row shows opinions regardless of the accuracy.

The left column contains opinions of users with blog-making experience. The middle column contains opinions of inexperienced users. The right column shows opinions regardless of the experience. The table leads to the following discussion:

(1) Expectations for multimedia blog creation

Users were satisfied with the system when high speech recognition accuracy was used regardless of their blog-making experience. Some users expected that the system could promote spread of multimedia contents with index keywords, even though few multimedia contents currently have indexes for retrieval.

(2) Robustness and tolerability for low accuracy

Users understood the results when low speech recognition accuracy was used because the multimedia content search is still fairly successful when keywords are accurately recognized, even though the total accuracy is not high. Users can appreciate the funny side of speech recognition errors and unexpected multimedia contents from PaPeRo's mistakes. However, as the errors do not always lead to amusing results, an edit interface should be equipped to improve keywords, illustrations and the total blog page layout.

(3) More expectations of dialogue with PaPeRo

Users would like to more enjoy themselves with PaPeRo, regardless of the speech recognition accuracy. They expect PaPeRo to give them more information, such as track-back and comments, based on dialogue history. As PaPeRo stores all the messages in himself, he has the ability to generate more sophisticated comments and track-back messages with users. Also, when the dialogue scenario is improved, he can ask the users some encouraging questions to make their blog more interesting and attractive while recording their video messages.

5 Conclusion

We developed a multimedia blog creation system using Japanese dialogue with an intelligent robot. The system was developed on PaPeRo for creating travel blogs and was evaluated by 10 users. Results showed that the system was effective for inexperienced and experienced users. The system enabled easy and simple creation of multimedia-rich blogs, while enabling users the pleasure of chatting with PaPeRo. We plan to improve the system by supporting the edit interface and enhancing the dialogue scenario so that users can enjoy themselves with more sophisticated and complex interaction with PaPeRo.

Table 2. Survey of Users' Opinions

		Blog-making experience		
		Experienced	Inexperienced	Either
Speech recognition accuracy	High	-This system makes multimedia contents more searchable on the internet.	-I would like to create blogs with PaPeRo.	-Easy to create blog only by chatting. -PaPeRo's comments are nice.
	Low	-Keywords, searched contents, and the total lay-out of blogs should be edited.	-Searched contents are good. -Even unexpectedly searched contents because of recognition errors are funny.	-PaPeRo could be allowed for his mistake. - Unexpected texts tempt users to play the video.
	Either	-PaPeRo's track-back is wanted as well as more dialogue variation.	-PaPeRo should talk on reasons of his choosing a song.	-PaPeRo should consider a history of recorded messages and his comments.

References

Yoshihiro Fujita. 2002. Personal Robot PaPeRo. *Journal of Robotics and Mechatronics,* 14(1): 60–63.

Takahiro Ikeda, at al. 2005. Speech-Activated Text Retrieval System for Cellular Phones with Web Browsing Capability. In *Proceedings of PACLIC19*, 265–272.

Ryosuke Isotani, et al. 2002. An Automatic Speech Translation System on PDAs for Travel Conversation. In *Proceedings of ICMI2002*, 211–216.

Ryosuke Isotani, et al. 2005. Basic Technologies for Spontaneous Speech Recognition and Its Applications. In *IPSJ-SIGNL*, 2005-NL-169, 209–116 (in Japanese).

Akira Nakamura (ed.). 1993. *Kanjo hyogen jiten (Emotional Expressions Dictionary).* Tokyodo Shuppan, Tokyo (in Japanese).

Stephen E. Robertson, et al. 1995. Okapi at TREC-3. In *Proceedings of TREC-3*, 109–126.

Koichi Shinoda and et al. 2002. Efficient Reduction of Gaussian Components Using MDL Criterion for HMM-based Speech Recognition, In *Proceedings of ICASSP-2002*, 869–872.

Takao Watanabe, et al. 1995. High Speed Speech Recognition Using Tree-Structured Probability Density Function. In *Proceedings of ICASSP-1995*, 556–559.

SemTAG: a platform for specifying Tree Adjoining Grammars and performing TAG-based Semantic Construction

Claire Gardent
CNRS / LORIA
Campus scientifique - BP 259
54 506 Vandœuvre-Lès-Nancy CEDEX
France
Claire.Gardent@loria.fr

Yannick Parmentier
INRIA / LORIA - Nancy Université
Campus scientifique - BP 259
54 506 Vandœuvre-Lès-Nancy CEDEX
France
Yannick.Parmentier@loria.fr

Abstract

In this paper, we introduce SEMTAG, a free and open software architecture for the development of Tree Adjoining Grammars integrating a compositional semantics. SEMTAG differs from XTAG in two main ways. First, it provides an expressive grammar formalism and compiler for factorising and specifying TAGs. Second, it supports semantic construction.

1 Introduction

Over the last decade, many of the main grammatical frameworks used in computational linguistics were extended to support semantic construction (i.e., the computation of a meaning representation from syntax and word meanings). Thus, the HPSG ERG grammar for English was extended to output minimal recursive structures as semantic representations for sentences (Copestake and Flickinger, 2000); the LFG (Lexical Functional Grammar) grammars to output lambda terms (Dalrymple, 1999); and Clark and Curran's CCG (Combinatory Categorial Grammar) based statistical parser was linked to a semantic construction module allowing for the derivation of Discourse Representation Structures (Bos et al., 2004).

For Tree Adjoining Grammar (TAG) on the other hand, there exists to date no computational framework which supports semantic construction. In this demo, we present SEMTAG, a free and open software architecture that supports TAG based semantic construction.

The structure of the paper is as follows. First, we briefly introduce the syntactic and semantic formalisms that are being handled (section 2). Second, we situate our approach with respect to other possible ways of doing TAG based semantic construction (section 3). Third, we show how XMG, the linguistic formalism used to specify the grammar (section 4) differs from existing computational frameworks for specifying a TAG and in particular, how it supports the integration of semantic information. Finally, section 5 focuses on the semantic construction module and reports on the coverage of SEMFRAG, a core TAG for French including both syntactic and semantic information.

2 Linguistic formalisms

We start by briefly introducing the syntactic and semantic formalisms assumed by SEMTAG namely, Feature-Based Lexicalised Tree Adjoining Grammar and L_U.

Tree Adjoining Grammars (TAG) TAG is a tree rewriting system (Joshi and Schabes, 1997). A TAG is composed of (i) two tree sets (a set of initial trees and a set of auxiliary trees) and (ii) two rewriting operations (substitution and adjunction). Furthermore, in a Lexicalised TAG, each tree has at least one leaf which is a terminal.

Initial trees are trees where leaf-nodes are labelled either by a terminal symbol or by a non-terminal symbol marked for substitution (↓). Auxiliary trees are trees where a leaf-node has the same label as the root node and is marked for adjunction ($\star$). This leaf-node is called a *foot* node.

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 13–16,
Prague, June 2007. ©2007 Association for Computational Linguistics

Further, substitution corresponds to the insertion of an *elementary* tree t_1 into a tree t_2 at a frontier node having the same label as the root node of t_1. Adjunction corresponds to the insertion of an *auxiliary* tree t_1 into a tree t_2 at an inner node having the same label as the root and foot nodes of t_1.

In a Feature-Based TAG, the nodes of the trees are labelled with two feature structures called top and bot. Derivation leads to unification on these nodes as follows. Given a substitution, the top feature structures of the merged nodes are unified. Given an adjunction, (i) the top feature structure of the inner node receiving the adjunction and of the root node of the inserted tree are unified, and (ii) the bot feature structures of the inner node receiving the adjunction and of the foot node of the inserted tree are unified. At the end of a derivation, the *top* and *bot* feature structures of each node in a derived tree are unified.

Semantics (L$_U$). The semantic representation language we use is a unification-based extension of the PLU language (Bos, 1995). L_U is defined as follows. Let H be a set of *hole* constants, L_c the set of *label* constants, and L_v the set of *label* variables. Let I_c (resp. I_v) be the set of individual constants (resp. variables), let R be a set of n-ary relations over $I_c \cup I_v \cup H$, and let $\geq$ be a relation over $H \cup L_c$ called the *scope-over* relation. Given $l \in L_c \cup L_v$, $h \in H$, $i_1, \ldots, i_n \in I_v \cup I_c \cup H$, and $R^n \in R$, we have:

1. $l : R^n(i_1, \ldots, i_n)$ is a L_U formula.
2. $h \geq l$ is a L_U formula.
3. ϕ, ψ is L_U formula iff both ϕ and ψ are L_U formulas.
4. Nothing else is a L_U formula.

In short, L_U is a flat (i.e., non recursive) version of first-order predicate logic in which scope may be underspecified and variables can be unification variables[1].

3 TAG based semantic construction

Semantic construction can be performed either during or after derivation of a sentence syntactic structure. In the first approach, syntactic structure and semantic representations are built simultaneously. This is the approach sketched by Montague and adopted e.g., in the HPSG ERG and in synchronous TAG (Nesson and Shieber, 2006). In the second approach, semantic construction proceeds from the syntactic structure of a complete sentence, from a lexicon associating each word with a semantic representation and from a set of semantic rules specifying how syntactic combinations relate to semantic composition. This is the approach adopted for instance, in the LFG glue semantic framework, in the CCG approach and in the approaches to TAG-based semantic construction that are based on the TAG derivation tree.

SEMTAG implements a hybrid approach to semantic construction where (i) semantic construction proceeds after derivation and (ii) the semantic lexicon is extracted from a TAG which simultaneously specifies syntax and semantics. In this approach (Gardent and Kallmeyer, 2003), the TAG used integrates syntactic and semantic information as follows. Each elementary tree is associated with a formula of L$_U$ representing its meaning. Importantly, the meaning representations of semantic functors include unification variables that are shared with specific feature values occurring in the associated elementary trees. For instance in figure 1, the variables x and y appear both in the semantic representation associated with the tree for *aime* (love) and in the tree itself.

Given such a TAG, the semantics of a tree t derived from combining the elementary trees $t_1, \ldots, t_n$ is the union of the semantics of $t_1, \ldots, t_n$ modulo the unifications that results from deriving that tree. For instance, given the sentence *Jean aime vraiment Marie* (*John really loves Mary*) whose TAG derivation is given in figure 1, the union of the semantics of the elementary trees used to derived the sentence tree is:

$$l_0 : jean(j),\ l_1 : aime(x, y),\ l_2 : vraiment(h_0),\\ l_s \leq h_0,\ l_3 : marie(m)$$

The unifications imposed by the derivations are:

$$\{x \rightarrow j, y \rightarrow m, l_s \rightarrow l_1\}$$

Hence the final semantics of the sentence *Jean aime vraiment Marie* is:

$$l_0 : jean(j),\ l_1 : aime(j, m),\ l_2 : vraiment(h_0),\\ l_1 \leq h_0,\ l_3 : marie(m)$$

[1] For mode details on L$_U$, see (Gardent and Kallmeyer, 2003).

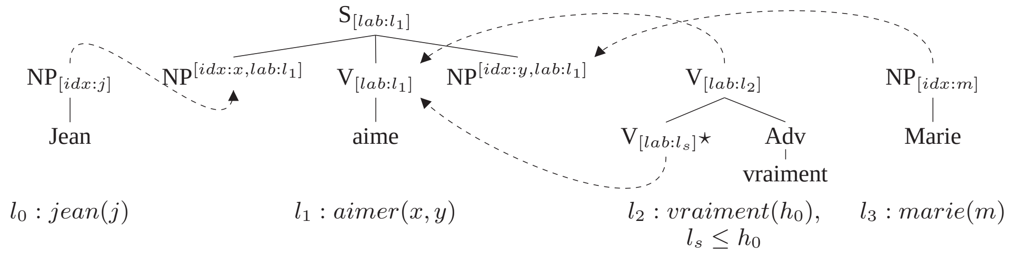


Figure 1: Derivation of "Jean aime vraiment Marie"

As shown in (Gardent and Parmentier, 2005), semantic construction can be performed either during or after derivation. However, performing semantic construction after derivation preserves modularity (changes to the semantics do not affect syntactic parsing) and allows the grammar used to remain within TAG (the grammar need contain neither an infinite set of variables nor recursive feature structures). Moreover, it means that standard TAG parsers can be used (if semantic construction was done during derivation, the parser would have to be adapted to handle the association of each elementary tree with a semantic representation). Hence in SEMTAG, semantic construction is performed after derivation. Section 5 gives more detail about this process.

4 The XMG formalism and compiler

SEMTAG makes available to the linguist a formalism (XMG) designed to facilitate the specification of tree based grammars integrating a semantic dimension. XMG differs from similar proposals (Xia et al., 1998) in three main ways (Duchier et al., 2004). First it supports the description of both syntax and semantics. Specifically, it permits associating each elementary tree with an L_U formula. Second, XMG provides an expressive formalism in which to *factorise* and *combine* the recurring tree fragments shared by several TAG elementary trees. Third, XMG provides a sophisticated treatment of variables which *inter alia*, supports variable sharing between semantic representation and syntactic tree. This sharing is implemented by means of so-called *interfaces* i.e., feature structures that are associated with a given (syntactic or semantic) fragment and whose scope is global to several fragments of the grammar specification.

To specify the syntax / semantics interface sketched in section 5, XMG is used as follows :

1. The elementary tree of a semantic functor is defined as the conjunction of its spine (the projection of its syntactic head) with the tree fragments describing each of its arguments. For instance, in figure 2, the tree for an intransitive verb is defined as the conjunction of the tree fragment for its spine (Active) with the tree fragment for (a canonical realisation of) its subject argument (Subject).

2. In the tree fragments representing the different syntactic realizations (canonical, extracted, etc.) of a given grammatical function, the node representing the argument (e.g., the subject) is labelled with an *idx* feature whose value is shared with a *GFidx* feature in the interface (where GF is the grammatical function).

3. Semantic representations are encapsulated as fragments where the semantic arguments are variables shared with the interface. For instance, the i^{th} argument of a semantic relation is associated with the *argI* interface feature.

4. Finally, the mapping between grammatical functions and thematic roles is specified when conjoining an elementary tree fragment with a semantic representation. For instance, in figure 2[2], the interface unifies the value of *arg1* (the thematic role) with that of *subjIdx* (a grammatical function) thereby specifying that the subject argument provides the value of the first semantic argument.

5 Semantic construction

As mentioned above, SEMTAG performs semantic construction after derivation. More specifically, semantic construction is supported by the following 3-step process:

[2] The interfaces are represented using gray boxes.

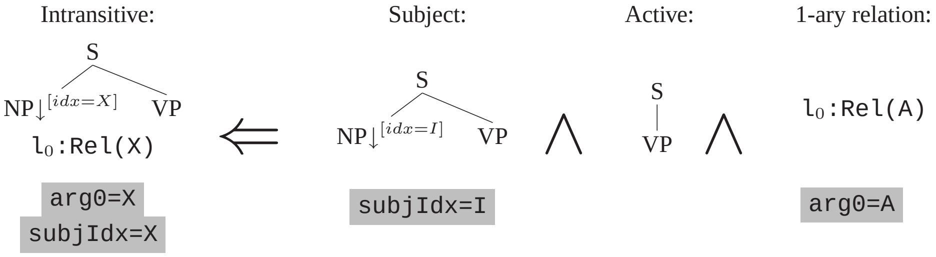


Figure 2: Syntax / semantics interface within the metagrammar.

1. First, we extract from the TAG generated by XMG (i) a purely syntactic TAG $\mathcal{G}'$, and (ii) a purely semantic TAG $\mathcal{G}''$ [3] A purely syntactic (resp. semantic) Tag is a TAG whose features are purely syntactic (resp. semantic) – in other words, $\mathcal{G}''$ is a TAG with no semantic features whilst $\mathcal{G}''$ is a TAG with only semantic features. Entries of $\mathcal{G}'$ and $\mathcal{G}''$ are indexed using the same key.

2. We generate a tabular syntactic parser for $\mathcal{G}'$ using the DyALog system of (de la Clergerie, 2005). This parser is then used to compute the derivation forest for the input sentence.

3. A semantic construction algorithm is applied to the derivation forest. In essence, this algorithm retrieves from the semantic TAG $\mathcal{G}''$ the semantic trees involved in the derivation(s) and performs on these the unifications prescribed by the derivation.

SEMTAG has been used to specify a core TAG for French, called SemFRag. This grammar is currently under evaluation on the *Test Suite for Natural Language Processing* in terms of syntactic coverage, semantic coverage and semantic ambiguity. For a test-suite containing 1495 sentences, 62.88 % of the sentences are syntactically parsed, 61.27 % of the sentences are semantically parsed (*i.e.*, at least one semantic representation is computed), and the average semantic ambiguity (number of semantic representation per sentence) is 2.46.

SEMTAG is freely available at `http://trac.loria.fr/~semtag`.

References

J. Bos, S. Clark, M. Steedman, J. R. Curran, and J. Hockenmaier. 2004. Wide-coverage semantic representations from a ccg parser. In *Proceedings of the 20th COLING*, Geneva, Switzerland.

J. Bos. 1995. Predicate Logic Unplugged. In *Proceedings of the tenth Amsterdam Colloquium, Amsterdam*.

A. Copestake and D. Flickinger. 2000. An open-source grammar development environment and broad-coverage english grammar using hpsg. In *Proceedings of LREC*, Athens, Greece.

Mary Dalrymple, editor. 1999. *Semantics and Syntax in Lexical Functional Grammar*. MIT Press.

E. de la Clergerie. 2005. DyALog: a tabular logic programming based environment for NLP. In *Proceedings of CSLP'05*, Barcelona.

D. Duchier, J. Le Roux, and Y. Parmentier. 2004. The Metagrammar Compiler: An NLP Application with a Multi-paradigm Architecture. In *Proceedings of MOZ'2004*, Charleroi.

C. Gardent and L. Kallmeyer. 2003. Semantic construction in FTAG. In *Proceedings of EACL'03, Budapest*.

C. Gardent and Y. Parmentier. 2005. Large scale semantic construction for tree adjoining grammars. In *Proceedings of LACL05, Bordeaux, France*.

A. Joshi and Y. Schabes. 1997. Tree-adjoining grammars. In G. Rozenberg and A. Salomaa, editors, *Handbook of Formal Languages*, volume 3, pages 69 – 124. Springer, Berlin, New York.

Rebecca Nesson and Stuart M. Shieber. 2006. Simpler TAG semantics through synchronization. In *Proceedings of the 11th Conference on Formal Grammar*, Malaga, Spain, 29–30 July.

F. Xia, M. Palmer, K. Vijay-Shanker, and J. Rosenzweig. 1998. Consistent grammar development using partial-tree descriptions for lexicalized tree adjoining grammar. *Proceedings of TAG+4*.

[3] As (Nesson and Shieber, 2006) indicates, this extraction in fact makes the resulting system a special case of synchronous TAG where the semantic trees are isomorphic to the syntactic trees and unification variables across the syntactic and semantic components are interpreted as synchronous links.

System Demonstration of On-Demand Information Extraction

Satoshi Sekine
New York University
715 Broadway, 7th floor
New York, NY 10003 USA
sekine@cs.nyu.edu

Akira Oda [1)]
Toyohashi University of Technology
1-1 Hibarigaoka, Tenpaku-cho,
Toyohashi, Aichi 441-3580 Japan
oda@ss.ics.tut.ac.jp

Abstract

In this paper, we will describe ODIE, the On-Demand Information Extraction system. Given a user's query, the system will produce tables of the salient information about the topic in structured form. It produces the tables in less than one minute without any knowledge engineering by hand, i.e. pattern creation or paraphrase knowledge creation, which was the largest obstacle in traditional IE. This demonstration is based on the idea and technologies reported in (Sekine 06). A substantial speed-up over the previous system (which required about 15 minutes to analyze one year of newspaper) was achieved through a new approach to handling pattern candidates; now less than one minute is required when using 11 years of newspaper corpus. In addition, functionality was added to facilitate investigation of the extracted information.

1 Introduction

The goal of information extraction (IE) is to extract information about events in structured form from unstructured texts. In traditional IE, a great deal of knowledge for the systems must be coded by hand in advance. For example, in the later MUC evaluations, system developers spent one month for the knowledge engineering to customize the system to the given test topic. Improving portability is necessary to make Information Extraction technology useful for real users and, we believe, lead to a breakthrough for the application of the technology.

Sekine (Sekine 06) proposed 'On-demand information extraction (ODIE)': a system which *automatically identifies the most salient structures and extracts the information on the topic the user demands*. This new IE paradigm becomes feasible due to recent developments in machine learning for NLP, in particular unsupervised learning methods, and is created on top of a range of basic language analysis tools, including POS taggers, dependency analyzers, and extended Named Entity taggers. This paper describes the demonstration system of the new IE paradigm, which incorporates some new ideas to make the system practical.

2 Algorithm Overview

We will present an overview of the algorithm in this section. The details can be found in (Sekine 06).

The basic functionality of the system is the following. The user types a query / topic description in keywords (for example, "merge, acquire, purchase"). Then tables will be created automatically while the user is waiting, rather than in a month of human labor. These tables are expected to show information about the salient relations for the topic.

There are six major components in the system.

1) IR system: Based on the query given by the user, it retrieves relevant documents from the document database. We used a simple TF/IDF IR system we developed.

2) Pattern discovery: The texts are analyzed using a POS tagger, a dependency analyzer and an Extended Named Entity (ENE) tagger, which will be explained in (5). Then sub-trees of dependency trees which are relatively frequent in the retrieved documents compared to the entire corpus are identified. The sub-trees to be used must satisfy some restrictions, including having

[1)] This work was conducted when the first author was a junior research scientist at New York University.

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 17–20,
Prague, June 2007.

between 2 and 6 nodes, having a predicate or nominalization as the head of the sub-tree, and having at least one NE. We introduced upper and lower frequency bounds for the sub-trees to be used, as we found the medium frequency sub-trees to be the most useful and least noisy. We compute a score for each pattern based on its frequency in the retrieved documents and in the entire collection. The top scoring sub-trees will be called *patterns*, which are expected to indicate salient relationships of the topic and which will be used in the later components. We pre-compute such information as much as possible in order to enable usably prompt response to queries.

3) <u>Paraphrase discovery</u>: In order to find semantic relationships between patterns, i.e. to find patterns which should be used to build the same table, we use lexical knowledge such as WordNet and paraphrase discovery techniques. The paraphrase discovery was conducted off-line and created a paraphrase knowledge base.

4) <u>Table construction</u>: In this component, the patterns created in (2) are linked based on the paraphrase knowledge base created by (3), producing sets of patterns which are semantically equivalent. Once the sets of patterns are created, these patterns are applied to the documents retrieved by the IR system (1). The matched patterns pull out the entity instances from the sentences and these entities are aligned to build the final tables.

5) <u>Extended NE tagger</u>: Most of the participants in events are likely to be Named Entities. However, the traditional NE categories are not sufficient to cover most participants of various events. For example, the standard MUC's 7 NE categories (i.e. person, location, organization, percent, money, time and date) miss product names (e.g. Windows XP, Boeing 747), event names (Olympics, World War II), numerical expressions other than monetary expressions, etc. We used the Extended NE with 140 categories and a tagger developed for these categories.

3 Speed-enhancing technology

The largest computational load in this system is the extraction and scoring of the topic-relevant sub-trees. In the previous system, 1,000 top-scoring sub-trees are extracted from all possible (on the order of hundreds of thousands) sub-trees in the top 200 relevant articles. This computation took about 14 minutes out of the total 15 minutes of the entire process. The difficulty is that the set of top articles is not predictable, as the input is arbitrary and hence the list of sub-trees is not predictable, too. Although a state-of-the-art tree mining algorithm (Abe et al. 02) was used, the computation is still impracticable for a real system.

The solution we propose in this paper is to precompute all possibly useful sub-trees in order to reduce runtime. We enumerate all possible sub-trees in the entire corpus and store them in a database with frequency and location information. To reduce the size of the database, we filter the patterns, keeping only those satisfying the constraints on frequency and existence of predicate and named entities. However, it is still a big challenge, because in this system, we use 11 years of newspaper (AQUAINT corpus, with duplicate articles removed) instead of the one year of newspaper (New York Times 95) used in the previous system. With this idea, the response time of the demonstration system is reduced significantly.

The statistics of the corpus and sub-trees are as follows. The entire corpus includes 1,031,124 articles and 24,953,026 sentences. The frequency thresholds for sub-trees to be used is set to more than 10 and less than 10,000; i.e. sub-trees of those frequencies in the corpus are expected to contain most of the salient relationships with minimum noise. The sub-trees with frequency less than 11 account for a very large portion of the data; 97.5% of types and 66.3% of instances, as shown in Table 1. The sub-trees of frequency of 10,001 or more are relatively small; only 76 kinds and only 2.5% of the instances.

Frequency	10,001 or more	10,000-11	10 or less
# of type	76	975,269	38,158,887
	~0.0%	2.5%	97.5%
# of instance	2,313,347	29,257,437	62,097,271
	2.5%	31.2%	66.3%

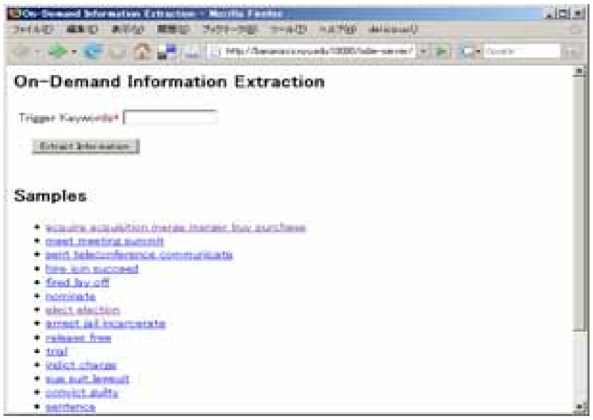
Table 1. Frequency of sub-trees

We assign ID numbers to all 1 million sub-trees and 25 million sentences and those are mutually linked in a database. Also, 60 million NE occurrences in the sub-trees are identified and linked to the sub-tree and sentence IDs. In the process, the sentences found by the IR component are identified. Then the sub-trees linked to those sentences are gathered and the scores are calculated. Those processes can be done by manipulation of the database in a very short time. The top sub-trees are used to create the output tables using NE occurrence IDs linked to the sub-trees and sentences.

4 A Demonstration

In this section, a simple demonstration scenario is presented with an example. Figure 1 shows the initial page. The user types in any keywords in the query box. This can be anything, but as a traditional IR system is used for the search, the keywords have to include expressions which are normally used in relevant documents. Examples of such keywords are "merge, acquisition, purchase", "meet, meeting, summit" and "elect, election", which were derived from ACE event types.

Then, normally within one minute, the system produces tables, such as those shown in Figure 2. All extracted tables are listed. Each table contains sentence ID, document ID and information extracted from the sentence. Some cells are empty if the information can't be extracted.



On-Demand Information Extraction

Trigger Keywords*

Extract Information

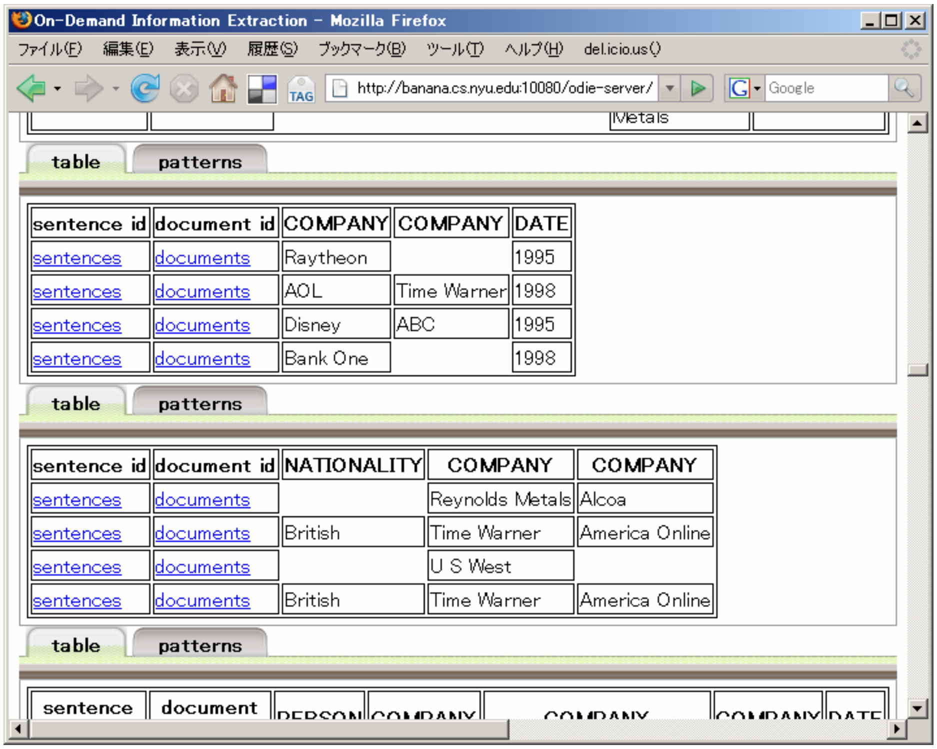
Samples

- acquire acquisition merge merger buy purchase
- meet meeting summit
- sent teleconference communicate
- hire join succeed
- fired lay off
- nominate
- elect election
- arrest jail incarcerate
- release free
- trial
- indict charge
- sue suit lawsuit
- convict guilty
- sentence

Figure 1. Screenshot of the initial page

5 Evaluation

The evaluation was conducted using scenarios based on 20 of the ACE event types. The accuracy of the extracted information was evaluated by judges for 100 rows selected at random. Of these rows, 66 were judged to be on target and correct. Another 10 were judged to be correct and related to the topic, but did not include the essential information of the topic. The remaining 24 included NE errors and totally irrelevant information (in some cases due to word sense ambiguity; e.g. "fine" weather vs."fine" as a financial penalty).


On-Demand Information Extraction - Mozilla Firefox
ファイル(F) 編集(E) 表示(V) 履歴(S) ブックマーク(B) ツール(T) ヘルプ(H) del.icio.us()
http://banana.cs.nyu.edu:10080/odie-server/
Google

Metals

table | patterns

sentence id	document id	COMPANY	COMPANY	DATE
sentences	documents	Raytheon		1995
sentences	documents	AOL	Time Warner	1998
sentences	documents	Disney	ABC	1995
sentences	documents	Bank One		1998

table | patterns

sentence id	document id	NATIONALITY	COMPANY	COMPANY
sentences	documents		Reynolds Metals	Alcoa
sentences	documents	British	Time Warner	America Online
sentences	documents		U S West	
sentences	documents	British	Time Warner	America Online

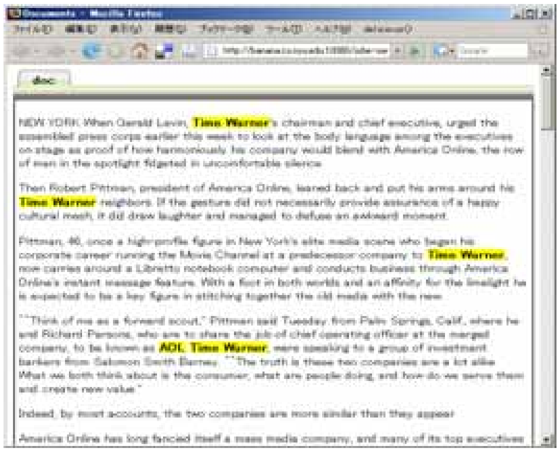
table | patterns

sentence	document	PERSON	COMPANY	COMPANY	COMPANY	DATE

Figure 2. Screenshot of produced tables

6 Other Functionality

Functionality is provided to facilitate the user's access to the extracted information. Figure 3 shows a screenshot of the document from which the information was extracted. Also the patterns used to create each table can be found by clicking the tab "patterns" (shown in Figure 4). This could help the user to understand the nature of the table. The information includes the frequency of the pattern in the retrieved documents and in the entire corpus, and the pattern's score.



NEW YORK: When Gerald Levin, Time Warner's chairman and chief executive, urged the assembled press corps earlier this week to look at the body language among the executives on stage as proof of how harmoniously his company would blend with America Online, the row of men in the spotlight fidgeted in uncomfortable silence.

Then Robert Pittman, president of America Online, leaned back and put his arms around his Time Warner neighbors. If the gesture did not necessarily provide assurance of a happy cultural mesh, it did draw laughter and managed to defuse an awkward moment.

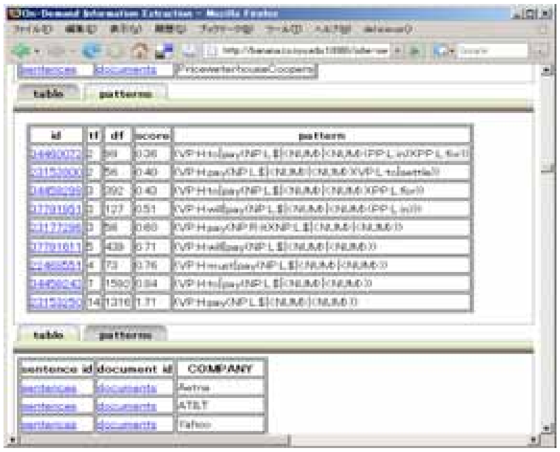
Pittman, 46, once a high-profile figure in New York's elite media scene who began his corporate career running the Movie Channel at a predecessor company to Time Warner, now carries around a Libretto notebook computer and conducts business through America Online's instant message feature. With a foot in both worlds and an affinity for the limelight he is expected to be a key figure in stitching together the old media with the new.

``Think of me as a forward scout,'' Pittman said Tuesday from Palm Springs, Calif., where he and Richard Parsons, who are to share the job of chief operating officer at the merged company, to be known as AOL Time Warner, were speaking to a group of investment bankers from Salomon Smith Barney. ``The truth is these two companies are a lot alike. What we both think about is the consumer, what are people doing, and how do we serve them and create new value.''

Indeed, by most accounts, the two companies are more similar than they appear.

America Online has long fancied itself a mass media company, and many of its top executives

Figure 3. Screenshot of document view



sentences | documents | PricewaterhouseCoopers

id	tf	df	score	pattern
04460072	2	98	0.36	(VP H:to|pay(NP L:$|<NUM>|<NUM>)(PP L:in)(PP L:for))
02153000	2	56	0.40	(VP H:pay(NP L:$|<NUM>|<NUM>)(VP L:to|settle))
04458299	3	392	0.40	(VP H:to|pay(NP L:$|<NUM>|<NUM>)(PP L:for))
07791901	3	127	0.51	(VP H:will|pay(NP L:$|<NUM>|<NUM>)(PP L:in))
02177296	3	56	0.60	(VP H:pay(NP R:it)(NP L:$|<NUM>|<NUM>))
07791611	5	439	0.71	(VP H:will|pay(NP L:$|<NUM>|<NUM>))
01466551	4	73	0.76	(VP H:must|pay(NP L:$|<NUM>|<NUM>))
04458242	7	1590	0.84	(VP H:to|pay(NP L:$|<NUM>|<NUM>))
02153250	14	1316	1.71	(VP H:pay(NP L:$|<NUM>|<NUM>))

sentence id	document id	COMPANY
sentences	documents	Aetna
sentences	documents	AT&T
sentences	documents	Yahoo

Figure 4. Screenshot of pattern information

7 Future Work

We demonstrated the On-Demand Information Extraction system, which provides usable response time for a large corpus. We still have several improvements to be made in the future. One is to include more advanced and accurate natural language technologies to improve the accuracy and coverage. For example, we did not use a coreference analyzer, and hence information which was expressed using pronouns or other anaphoric expressions can not be extracted. Also, more semantic knowledge including synonym, paraphrase or inference knowledge should be included. The output table has to be more clearly organized. In particular, we can't display role information as column headings. The keyword input requirement is very inconvenient. For good performance, the current system requires several keywords occurring in relevant documents; this is an obvious limitation. On the other hand, there are systems which don't need any user input to create the structured information (Banko et al. 07) (Shinyama and Sekine 06). The latter system tries to identify all possible structural relations from a large set of unstructured documents. However, the user's information needs are not predictable and the question of whether we can create structured information for all possible needs is still a big challenge.

Acknowledgements

This research was supported in part by the Defense Advanced Research Projects Agency as part of the Translingual Information Detection, Extraction and Summarization (TIDES) program, under Grant N66001-001-1-8917 from the Space and Naval Warfare Systems Center, San Diego, and by the National Science Foundation under Grant IIS-00325657. This paper does not necessarily reflect the position of the U.S. Government. We would like to thank our colleagues at New York University, who provided useful suggestions and discussions, including, Prof. Ralph Grishman and Mr. Yusuke Shinyama.

References

Kenji Abe, Shinji Kawasone, Tatsuya Asai, Hiroki Arimura and Setsuo Arikawa. 2002. "Optimized Substructure Discovery for Semi-structured Data". PKDD-02.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead and Oren Etzioni. 2007. "Open Information Extraction from Web". IJCAI-07.

Satoshi Sekine. 2006. "On-Demand Information Extraction". COLING-ACL-06.

Yusuke Shinyama and Satoshi Sekine, 2006. "Preemptive Information Extraction using Unrestricted Relation Discovery". HLT-NAACL-2006.

Multilingual Ontological Analysis of European Directives

Gianmaria Ajani
Dipartimento di Scienze Giuridiche
Università di Torino - Italy
gianmaria.ajani@unito.it

Guido Boella
Leonardo Lesmo
Alessandro Mazzei
Dipartimento di Informatica
Università di Torino - Italy
[guido|lesmo|mazzei]@di.unito.it

Piercarlo Rossi
Dipartimento di Studi per l'Impresa e il Territorio
Università del Piemonte Orientale - Italy
piercarlo.rossi@eco.unipmn.it

Abstract

This paper describes the main features of our tool called "Legal Taxonomy Syllabus". The system is an ontology based tool designed to annotate and recover multi-lingua legal information and build conceptual dictionaries on European Directives.

1 Introduction

The European union each year produces a large number of Union Directives (EUD), which are translated into each of the communitary languages. The EUD are sets of norms that have to be implemented by the national legislations. The problem of multilinguism in European legislation has recently been addressed by using linguistic and ontological tools, e.g. (Boer et al., 2003; Giguet and P.S., 2006; Després and Szulman, 2006). The management of EUD is particularly complex since the implementation of a EUD however not correspond to the straight transposition into a national law. An EUD is subject to further interpretation, and this process can lead to unexpected results. Comparative Law has studied in details the problematics concerning EUD and their complexities. On the other hand managing with appropriate tools this kind of complexity can facilitate the comparison and harmonization of national legislation (Boer et al., 2003). Based on this research, in this paper, we describe the tool for building multilingual conceptual dictionaries we developed for representing an analysing the terminology and concepts used in EUD.

The main assumptions of our methodology, motivated by studies in comparative law (Rossi and Vogel, 2004) and ontologies engineering (Klein, 2001), are the following ones: 1) Terms and concepts must be distinguished; for this purpose, we use lightweight ontologies, i.e. simple taxonomic structures of primitive or composite terms together with associated definitions. They are hardly axiomatized as the intended meaning of the terms used by the community is more or less known in advance by all members, and the ontology can be limited to those structural relationships among terms that are considered as relevant (Oberle, 2005)[1]. 2) We distinguish the ontology implicitly defined by EUD, the *EU level*, from the various national ontologies, the *national level*. Furthermore, each national legislation refers to a distinct national legal ontology. We do not assume that the transposition of an EUD introduces automatically in a national ontology the same concepts present at the EU level. 3) Corresponding concepts at the EU level and at the national level can be denoted by different terms in the same national language.

In this paper, we show how the Legal Taxonomy Syllabus (LTS) is used to build a dictionary of consumer law, to support the Uniform Terminology Project[2] (Rossi and Vogel, 2004). The structure of this paper is the following one. In Section 2 we stress two main problems which comparative law has raised concerning EUD and their transpositions. In Section 3 we describe how the methodology of the LTS allows to cope with these problems and finally in Section 4we give some conclusions.

[1] See http://cos.ontoware.org/

[2] http://www.uniformterminology.unito.it

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 21–24,
Prague, June 2007. ©2007 Association for Computational Linguistics

2 Terminological and conceptual misalignment

Comparative law has identified two key points in dealing with EUD, which makes more difficult dealing with the polysemy of legal terms: we call them the *terminological* and *conceptual misalignments*.

In the case of EUD (usually adopted for harmonising the laws of the Member States), the terminological matter is complicated by their necessity to be implemented by the national legislations. In order to have a precise transposition in a national law, a Directive may be subject to further interpretation. Thus, a same *legal concept* can be expressed in different ways in a Directive and in its implementing national law. The same legal concept in some language can be expressed in a different way in a EUD and in the national law implementing it. As a consequence we have a terminological misalignment. For example, the concept corresponding to the word *reasonably* in English, is translated into Italian as *ragionevolmente* in the EUD, and as *con ordinaria diligenza* into the transposition law.

In the EUD transposition laws a further problem arises from the different national *legal doctrines*. A legal concept expressed in an EUD may not be present in a national legal system. In this case we can talk about a conceptual misalignment. To make sense for the national lawyers' expectancies, the European legal terms have not only to be translated into a sound national terminology, but they need to be correctly detected when their meanings are to refer to EU legal concepts or when their meanings are similar to concepts which are known in the Member states. Consequently, the transposition of European law in the parochial legal framework of each Member state can lead to a set of distinct national legal doctrines, that are all different from the European one. In case of consumer contracts (like those concluded by the means of distance communication as in Directive 97/7/EC, Art. 4.2), the notion to provide in a *clear and comprehensible manner* some elements of the contract by the professionals to the consumers represents a specification of the information duties which are a pivotal principle of EU law. Despite the pairs of translation in the language versions of EU Directives (i.e., *klar und verständlich* in German - *clear and comprehensible* in English - *chiaro e comprensibile* in Italian), each legal term, when transposed in the national legal orders, is influenced by the conceptual filters of the lawyers' domestic legal thinking. So, *klar und verständlich* in the German system is considered by the German commentators referring to three different legal concepts: 1) the print or the writing of the information must be clear and legible (*gestaltung der information*), 2) the information must be intelligible by the consumer (*formulierung der information*), 3) the language of the information must be the national of consumer (*sprache der information*). In Italy, the judiciary tend to control more the formal features of the concepts 1 and 3, and less concept 2, while in England the main role has been played by the concept 2, though considered as plain style of language (not legal technical jargon) thanks to the historical influences of plain English movement in that country.

Note that this kind of problems identified in comparative law has a direct correspondence in the ontology theory. In particular Klein (Klein, 2001) has remarked that two particular forms of ontology mismatch are *terminological* and *conceptualization* ontological mismatch which straightforwardly correspond to our definitions of misalignments.

3 The methodology of the Legal Taxonomy Syllabus

A standard way to properly manage large multilingual lexical databases is to do a clear distinction among terms and their interlingual acceptions (or *axies*) (Sérasset, 1994; Lyding et al., 2006). In our system to properly manage terminological and conceptual misalignment we distinguish in the LTS project the notion of legal term from the notion of legal concept and we build a systematic classification based on this distinction. The basic idea in our system is that the conceptual backbone consists in a taxonomy of concepts (ontology) to which the terms can refer to express their meaning. One of the main points to keep in mind is that we do not assume the existence of a single taxonomy covering all languages. In fact, it has been convincingly argued that the different national systems may organize the concepts in different ways. For instance, the term *contract* corresponds to different concepts

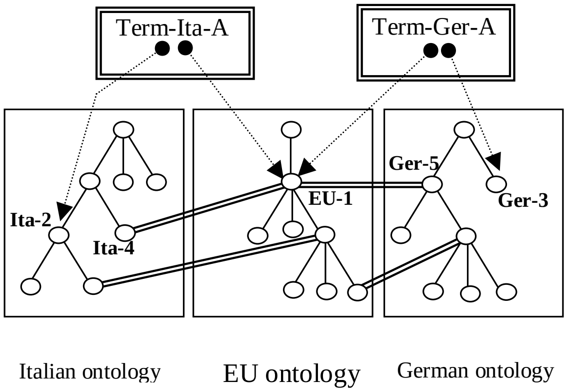


Figure 1: Relationship between ontologies and terms. The thick arcs represent the inter-ontology “association” link.

in common law and civil law, where it has the meaning of *bargain* and *agreement*, respectively (Sacco, 1999). In most complex instances, there are no homologous between terms-concepts such as *frutto civile* (legal fruit) and *income*, but respectively civil law and common law systems can achieve functionally same operational rules thanks to the functioning of the entire taxonomy of national legal concepts (Graziadei, 2004). Consequently, the LTS includes different ontologies, one for each involved national language plus one for the language of EU documents. Each language-specific ontology is related via a set of *association* links to the EU concepts, as shown in Fig. 1.

Although this picture is conform to intuition, in LTS it had to be enhanced in two directions. First, it must be observed that the various national ontologies have a reference language. This is not the case for the EU ontology. For instance, a given term in English could refer either to a concept in the UK ontology or to a concept in the EU ontology. In the first case, the term is used for referring to a concept in the national UK legal system, whilst in the second one, it is used to refer to a concept used in the European directives. This is one of the main advantages of LTS. For example *klar und verständlich* could refer both to concept `Ger-379` (a concept in the German Ontology) and to concept `EU-882` (a concept in the European ontology). This is the LTS solution for facing the possibility of a correspondence only partial between the meaning of a term has in the national system and the meaning of the same term in the translation of a EU directive. This feature enables the LTS to be more precise about what “translation” means. It puts at disposal a way for asserting that two terms are the translation of each other, but just in case those terms have been used in the translation of an EU directive: within LTS, we can talk about direct EU-translations of terms, but only about indirect national-system translations of terms. The situation enforced in LTS is depicted in Fig. 1, where it is represented that: The Italian term *Term-Ita-A* and the German term *Term-Ger-A* have been used as corresponding terms in the translation of an EU directive, as shown by the fact that both of them refer to the same EU-concept `EU-1`. In the Italian legal system, *Term-Ita-A* has the meaning `Ita-2`. In the German legal system, *Term-Ger-A* has the meaning `Ger-3`. The EU translations of the directive is correct insofar no terms exist in Italian and German that characterize precisely the concept `EU-1` in the two languages (i.e., the “associated” concepts `Ita-4` and `Ger-5` have no corresponding legal terms). A practical example of such a situation is reported in Fig. 2, where we can see that the ontologies include different types of arcs. Beyond the usual *is-a* (linking a category to its supercategory), there are also a *purpose* arc, which relates a concept to the legal principle motivating it, and *concerns*, which refers to a general relatedness. The dotted arcs represent the reference from terms to concepts. Some terms have links both to a National ontology and to the EU Ontology (in particular, *withdrawal* vs. *recesso* and *difesa del consumatore* vs. *consumer protection*).

The last item above is especially relevant: note that this configuration of arcs specifies that: 1) *withdrawal* and *recesso* have been used as equivalent terms (concept `EU-2`) in some European Directives (e.g., Directive 90/314/EEC). 2) In that context, the term involved an act having as purpose the some kind of protection of the consumer. 3) The terms used for referring to the latter are *consumer protection* in English and *difesa del consumatore* in Italian. 4) In the British legal system, however, not all *withdrawals* have this goal, but only a subtype of them, to which the code refers to as *cancellation* (concept `Eng-3`). 5) In the Italian legal system, the term *diritto di recesso* is ambiguous, since it can be used with reference either to something concerning

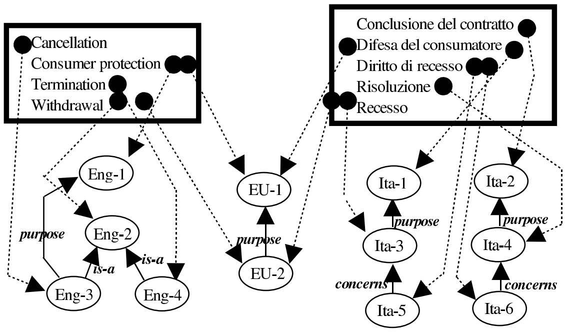


Figure 2: An example of interconnections among terms.

the *risoluzione* (concept `Ita-4`), or to something concerning the *recesso* proper (concept `Ita-3`).

Finally, it is possible to use the LTS to translate terms into different national systems via the concepts which they are transposition of at the European level. For instance suppose that we want to translate the legal term *credito al consumo* from Italian to German. In the LTS *credito al consumo* is associated to the national umeaning `Ita-175`. We find that `Ita-175` is the transposition of the European umeaning `EU-26` (*contratto di credito*). `EU-26` is associated to the German legal term *Kreditvertrag* at European level. Again, we find that the national German transposition of `EU-26` corresponds to the national umeaning `Ger-32` that is associated with the national legal term *Darlehensvertrag*. Then, by using the European ontology, we can translate the Italian legal term *credito al consumo* into the German legal term *Darlehensvertrag*.

4 Conclusions

In this paper we discuss some features of the LTS, a tool for building multilingual conceptual dictionaries for the EU law. The tool is based on lightweight ontologies to allow a distinction of concepts from terms. Distinct ontologies are built at the EU level and for each national language, to deal with polysemy and terminological and conceptual misalignment.

Many attempts have been done to use ontology in legal field, e.g. (Casanovas et al., 2005; Després and Szulman, 2006) and LOIS project (that is based on EuroWordNet project (Vossen et al., 1999), `http://www.loisproject.org`), but to our knowledge the LTS is the first attempt which starts from fine grained legal expertise on the EUD domain.

Future work is to study how the LTS can be used as a thesaurus for general EUD, even if the current domain is limited to consumer law.

References

A. Boer, T.M. van Engers, and R. Winkels. 2003. Using ontologies for comparing and harmonizing legislation. In *ICAIL*, pages 60–69.

P. Casanovas, N. Casellas, C. Tempich, D. Vrandecic, and R. Benjamins. 2005. OPJK modeling methodology. In *Proceedings of the ICAIL Workshop: LOAIT 2005*.

S. Després and S. Szulman. 2006. Merging of legal micro-ontologies from european directives. *Artificial Intelligence and Law*, ??:??–?? In press.

E. Giguet and P.S. 2006. Multilingual lexical database generation from parallel texts in 20 european languages with endogenous resources. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 271–278, July.

M. Graziadei. 2004. Tuttifrutti. In P. Birks and A. Pretto, editors, *Themes in Comparative Law*, pages –. Oxford University Press.

M. Klein. 2001. Combining and relating ontologies: an analysis of problems and solutions. In *Workshop on Ontologies and Information Sharing, IJCAI'01*, Seattle, USA.

V. Lyding, Elena Chiocchetti, G. Sérasset, and F. Brunet-Manquat. 2006. The LexALP information system: Term bank and corpus for multilingual legal terminology consolidated. In *Proc. of the Wokshop on Multilingual Language Resources and Interoperability, ACL06*, pages 25–31.

D. Oberle, editor. 2005. *Semantic Management of Middleware*. Springer Science+Business and Media.

P. Rossi and C. Vogel. 2004. Terms and concepts; towards a syllabus for european private law. *European Review of Private Law (ERPL)*, 12(2):293–300.

R. Sacco. 1999. Contract. *European Review of Private Law*, 2:237–240.

G. Sérasset. 1994. Interlingual lexical organization for multilingual lexical databases in NADIA. In *Proc. COLING94*, pages 278–282.

P. Vossen, W. Peters, and J. Gonzalo. 1999. Towards a universal index of meaning. In *Proc. ACL-99 Siglex Workshop*.

NICT-ATR Speech-to-Speech Translation System

Eiichiro Sumita **Tohru Shimizu** **Satoshi Nakamura**

National Institute of Information and Communications Technology
&
ATR Spoken Language Communication Research Laboratories
2-2-2 Hikaridai, Keihanna Science City, Kyoto 619-0288, Japan
eiichiro.sumita, tohru.shimizu & satoshi.nakamura@atr.jp

Abstract

This paper describes the latest version of speech-to-speech translation systems developed by the team of NICT-ATR for over twenty years. The system is now ready to be deployed for the travel domain. A new noise-suppression technique notably improves speech recognition performance. Corpus-based approaches of recognition, translation, and synthesis enable coverage of a wide variety of topics and portability to other languages.

1 Introduction

Speech recognition, speech synthesis, and machine translation research started about half a century ago. They have developed independently for a long time until speech-to-speech translation research was proposed in the 1980's. The feasibility of speech-to-speech translation was the focus of research at the beginning because each component was difficult to build and their integration seemed more difficult. After groundbreaking work for two decades, corpus-based speech and language processing technology have recently enabled the achievement of speech-to-speech translation that is usable in the real world.

This paper introduces (at ACL 2007) the state-of-the-art speech-to-speech translation system developed by NICT-ATR, Japan.

2 SPEECH-TO-SPEECH TRANSLATION SYSTEM

A speech-to-speech translation system is very large and complex. In this paper, we prefer to describe recent progress. Detailed information can be found in [1, 2, 3] and their references.

2.1 Speech recognition

To obtain a compact, accurate model from corpora with a limited size, we use MDL-SSS [4] and composite multi-class N-gram models [5] for acoustic and language modeling, respectively. MDL-SSS is an algorithm that automatically determines the appropriate number of parameters according to the size of the training data based on the Maximum Description Length (MDL) criterion. Japanese, English, and Chinese acoustic models were trained using the data from 4,200, 532, and 536 speakers, respectively. Furthermore, these models were adapted to several accents, e.g., US (the United States), AUS (Australia), and BRT (Britain) for English. A statistical language model was trained by using large-scale corpora (852 k sentences of Japanese, 710 k sentences of English, 510 k sentences of Chinese) drawn from the travel domain.

Robust speech recognition technology in noisy situations is an important issue for speech translation in real-world environments. An MMSE (Minimum mean square error) estimator for log Mel-spectral energy coefficients using a GMM (Gaussian Mixture Model) [6] is introduced for suppressing interference and noise and for attenuating reverberation.

Even when the acoustic and language models are trained well, environmental conditions such as variability of speakers, mismatches between the training and testing channels, and interference from environmental noise may cause recognition errors. These utterance recognition errors can be rejected by tagging them with a low confidence value. To do this we introduce generalized word

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 25–28,
Prague, June 2007.

posterior probability (GWPP)-based recognition error rejection for the post processing of the speech recognition [7, 8].

2.2 Machine translation

The translation modules are automatically constructed from large-scale corpora: (1) TATR, a phrase-based SMT module and (2) EM, a simple memory-based translation module. EM matches a given source sentence against the source language parts of translation examples. If an exact match is achieved, the corresponding target language sentence will be output. Otherwise, TATR is called up. In TATR, which is built within the framework of feature-based exponential models, we used the following five features: phrase translation probability from source to target; inverse phrase translation probability; lexical weighting probability from source to target; inverse lexical weighting probability; and phrase penalty.

Here, we touch on two approaches of TATR: novel word segmentation for Chinese, and language model adaptation.

We used a subword-based approach for word segmentation of Chinese [9]. This word segmentation is composed of three steps. The first is a dictionary-based step, similar to the word segmentation provided by LDC. The second is a subword-based IOB tagging step implemented by a CRF tagging model. The subword-based IOB tagging achieves a better segmentation than character-based IOB tagging. The third step is confidence-dependent disambiguation to combine the previous two results. The subword-based segmentation was evaluated with two different data from the Sighan Bakeoff and the NIST machine translation evaluation workshop. With the data of the second Sighan Bakeoff[1], our segmentation gave a higher F-score than the best published results. We also evaluated this segmentation in a translation scenario using the data of NIST translation evaluation[2] 2005, where its BLEU score[3] was 1.1% higher than that using the LDC-provided word segmentation.

The language model that is used plays an important role in SMT. The effectiveness of the language model is significant if the test data happen to have the same characteristics as those of the training data for the language models. However, this coincidence is rare in practice. To avoid this performance reduction, a topic adaptation technique is often used. We applied this adaptation technique to machine translation. For this purpose, a “topic” is defined as clusters of bilingual sentence pairs. In the decoding, for a source input sentence, f, a topic T is determined by maximizing P(f|T). To maximize P(f|T) we select cluster T that gives the highest probability for a given translation source sentence f. After the topic is found, a topic-dependent language model P(e|T) is used instead of P(e), the topic-independent language model. The topic-dependent language models were tested using IWSLT06 data[4]. Our approach improved the BLEU score between 1.1% and 1.4%. The paper of [10] presents a detailed description of this work.

2.3 Speech synthesis

An ATR speech synthesis engine called XIMERA was developed using large corpora (a 110-hour corpus of a Japanese male, a 60-hour corpus of a Japanese female, and a 20-hour corpus of a Chinese female). This corpus-based approach makes it possible to preserve the naturalness and personality of the speech without introducing signal processing to the speech segment [11]. XIMERA’s HMM (Hidden Markov Model)-based statistical prosody model is automatically trained, so it can generate a highly natural F0 pattern [12]. In addition, the cost function for segment selection has been optimized based on perceptual experiments, thereby improving the naturalness of the selected segments [13].

3 EVALUATION

3.1 Speech and language corpora

We have collected three kinds of speech and language corpora: BTEC (Basic Travel Expression Corpus), MAD (Machine Aided Dialog), and FED (Field Experiment Data) [14, 15, 16, and 17]. The BTEC Corpus includes parallel sentences in two languages composed of the kind of sentences one might find in a travel phrasebook. MAD is a dialog corpus collected using a speech-to-speech translation system. While the size of this corpus is relatively limited, the corpus is used for adaptation and

[1] http://sighan.cs.uchicago.edu/bakeoff2005/

[2] http://www.nist.gov/speech/tests/mt/mt05eval_official_results_release_20050801_v3.html

[3] http://www.nist.gov/speech/tests/mt/resources/scoring.htm

[4] http://www.slt.atr.jp/IWSLT2006/

evaluation. FED is a corpus collected in Kansai International Airport uttered by travelers using the airport.

3.2 Speech recognition system

The size of the vocabulary was about 35 k in canonical form and 50 k with pronunciation variations. Recognition results are shown in Table 1 for Japanese, English, and Chinese with a real-time factor[5] of 5. Although the speech recognition performance for dialog speech is worse than that for read speech, the utterance correctness excluding erroneous recognition output using GWPP [8] was greater than 83% in all cases.

		BTEC	MAD	FED
Characteristics		Read speech	Dialog speech (Office)	Dialog speech (Airport)
# of speakers		20	12	6
# of utterances		510	502	155
# of word tokens		4,035	5,682	1,108
Average length		7.9	11.3	7.1
Perplexity		18.9	23.2	36.2
Word accuracy	*Japanese*	94.9	92.9	91.0
	English	92.3	90.5	81.0
	Chinese	90.7	78.3	76.5
Utterance correctness	*All*	82.4	62.2	69.0
	Not rejected	87.1	83.9	91.4

Table 1 Evaluation of speech recognition

3.3 Machine Translation

The mechanical evaluation is shown, where there are sixteen reference translations. The performance is very high except for English-to-Chinese (Table 2).

	BLEU
Japanese-to-English	0.6998
English-to-Japanese	0.7496
Japanese-to-Chinese	0.6584
Chinese-to-Japanese	0.7400
English-to-Chinese	0.5520
Chinese-to-English	0.6581

Table 2 Mechanical evaluation of translation

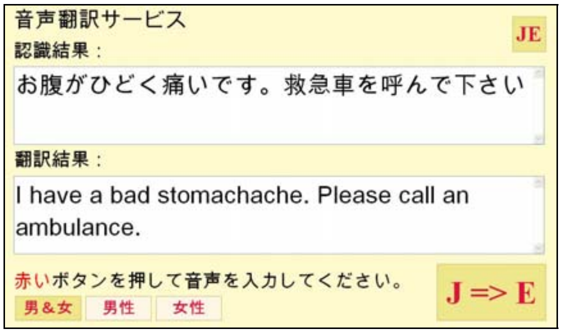
The translation outputs were ranked A (perfect), B (good), C (fair), or D (nonsense) by professional translators. The percentage of ranks is shown in Table 3. This is in accordance with the above BLEU score.

	A	AB	ABC
Japanese-to-English	78.4	86.3	92.2
English-to-Japanese	74.3	85.7	93.9
Japanese-to-Chinese	68.0	78.0	88.8
Chinese-to-Japanese	68.6	80.4	89.0
English-to-Chinese	52.5	67.1	79.4
Chinese-to-English	68.0	77.3	86.3

Table 3 Human Evaluation of translation

4 System presented at ACL 2007

The system works well in a noisy environment and translation can be performed for any combination of Japanese, English, and Chinese languages. The display of the current speech-to-speech translation system is shown below.



お腹がひどく痛いです。救急車を呼んで下さい

I have a bad stomachache. Please call an ambulance.

赤いボタンを押して音声を入力してください。

Figure 1 Japanese-to-English Display of NICT-ATR Speech-to-Speech Translation System

5 CONCLUSION

This paper presented a speech-to-speech translation system that has been developed by NICT-ATR for two decades. Various techniques, such as noise suppression and corpus-based modeling for both speech processing and machine translation achieve robustness and portability.

The evaluation has demonstrated that our system is both effective and useful in a real-world environment.

[5] The real time factor is the ratio to an utterance time.

References

[1] S. Nakamura, K. Markov, H. Nakaiwa, G. Kikui, H. Kawai, T. Jitsuhiro, J. Zhang, H. Yamamoto, E. Sumita, and S. Yamamoto. The ATR multilingual speech-to-speech translation system. IEEE Trans. on Audio, Speech, and Language Processing, 14, No. 2:365–376, 2006.

[2] T. Shimizu, Y. Ashikari, E. Sumita, H. Kashioka, and S. Nakamura, "Development of client-server speech translation system on a multi-lingual speech communication platform," Proc. of the International Workshop on Spoken Language Translation, pp. 213-216, Kyoto, Japan, 2006.

[3] R. Zhang, H. Yamamoto, M. Paul, H. Okuma, K. Yasuda, Y. Lepage, E. Denoual, D. Mochihashi, A. Finch, and E. Sumita, "The NiCT-ATR Statistical Machine Translation System for the IWSLT 2006 Evaluation," Proc. of the International Workshop on Spoken Language Translation, pp. 83-90, Kyoto, Japan , 2006.

[4] T. Jitsuhiro, T. Matsui, and S. Nakamura. Automatic generation of non-uniform context-dependent HMM topologies based on the MDL criterion. In Proc. of Eurospeech, pages 2721–2724, 2003.

[5] H. Yamamoto, S. Isogai, and Y. Sagisaka. Multi-class composite N-gram language model. Speech Communication, 41:369–379, 2003.

[6] M. Fujimoto and Y. Ariki. Combination of temporal domain SVD based speech enhancement and GMM based speech estimation for ASR in noise - evaluation on the AURORA II database and tasks. In Proc. of Eurospeech, pages 1781–1784, 2003.

[7] F. K. Soong, W. K. Lo, and S. Nakamura. Optimal acoustic and language model weight for minimizing word verification errors. In Proc. of ICSLP, pages 441–444, 2004

[8] W. K. Lo and F. K. Soong. Generalized posterior probability for minimum error verification of recognized sentences. In Proc. of ICASSP, pages 85–88, 2005.

[9] R. Zhang, G. Kikui, and E. Sumita, "Subword-based tagging by conditional random fields for Chinese word segmentation," in Companion volume to the proceedings of the North American chapter of the Association for Computational Linguistics (NAACL), 2006, pp. 193–196.

[10] H. Yamamoto and E. Sumita, "Online language model task adaptation for statistical machine translation (in Japanese)," in FIT2006, Fukuoka, Japan, 2006, pp. 131–134.

[11] H. Kawai, T. Toda, J. Ni, and M. Tsuzaki. XIMERA: A new TTS from ATR based on corpus-based technologies. In Proc. of 5th ISCA Speech Synthesis Workshop, 2004.

[12] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura. Speech parameter generation algorithms for HMM-based speech synthesis. In Proc. of ICASSP, pages 1215–1218, 2000.

[13] T. Toda, H. Kawai, and M. Tsuzaki. Optimizing sub-cost functions for segment selection based on perceptual evaluation in concatenative speech synthesis. In Proc. of ICASSP, pages 657–660, 2004.

[14] T. Takezawa and G. Kikui. Collecting machine – translation-aided bilingual dialogs for corpus-based speech translation. In Proc. of Eurospeech, pages 2757–2760, 2003.

[15] G. Kikui, E. Sumita, T. Takezawa, and S. Yamamoto. Creating corpora for speech-to-speech translation. In Proc. Of Eurospeech, pages 381–384, 2003.

[16] T. Takezawa and G. Kikui. A comparative study on human communication behaviors and linguistic characteristics for speech-to-speech translation. In Proc. of LREC, pages 1589–1592, 2004.

[17] G. Kikui, T. Takezawa, M. Mizushima, S. Yamamoto, Y. Sasaki, H. Kawai, and S. Nakamura. Monitor experiments of ATR speech-to-speech translation system. In Proc. of Autumn Meeting of the Acoustical Society of Japan, pages 1–7–10, 2005, in Japanese.

zipfR: Word Frequency Distributions in R

Stefan Evert
IKW (University of Osnabrück)
Albrechtstr. 28
49069 Osnabrück, Germany
stefan.evert@uos.de

Marco Baroni
CIMeC (University of Trento)
C.so Bettini 31
38068 Rovereto, Italy
marco.baroni@unitn.it

Abstract

We introduce the *zipfR* package, a powerful and user-friendly open-source tool for LNRE modeling of word frequency distributions in the R statistical environment. We give some background on LNRE models, discuss related software and the motivation for the toolkit, describe the implementation, and conclude with a complete sample session showing a typical LNRE analysis.

1 Introduction

As has been known at least since the seminal work of Zipf (1949), words and other type-rich linguistic populations are characterized by the fact that even the largest samples (corpora) do not contain instances of all types in the population. Consequently, the number and distribution of types in the available sample are not reliable estimators of the number and distribution of types in the population. Large-Number-of-Rare-Events (LNRE) models (Baayen, 2001) are a class of specialized statistical models that estimate the distribution of occurrence probabilities in such type-rich linguistic populations from our limited samples.

LNRE models have applications in many branches of linguistics and NLP. A typical use case is to predict the number of different types (the *vocabulary size*) in a larger sample or the whole population, based on the smaller sample available to the researcher. For example, one could use LNRE models to infer how many words a 5-year-old child knows in total, given a sample of her writing. LNRE models can also be used to quantify the relative productivity of two morphological processes (as illustrated below) or of two rival syntactic constructions by looking at their vocabulary growth rate as sample size increases. Practical NLP applications include making informed guesses about type counts in very large data sets (e.g., *How many typos are there on the Internet?*) and determining the "lexical richness" of texts belonging to different genres. Last but not least, LNRE models play an important role as a population model for Bayesian inference and Good-Turing frequency smoothing (Good, 1953).

However, with a few notable exceptions (such as the work by Baayen on morphological productivity), LNRE models are rarely if ever employed in linguistic research and NLP applications. We believe that this has to be attributed, at least in part, to the lack of easy-to-use but sophisticated LNRE modeling tools that are reliable and robust, scale up to large data sets, and can easily be integrated into the workflow of an experiment or application. We have developed the *zipfR* toolkit in order to remedy this situation.

2 LNRE models

In the field of LNRE modeling, we are not interested in the frequencies or probabilities of individual word types (or types of other linguistic units), but rather in the *distribution* of such frequencies (in a sample) and probabilities (in the population). Consequently, the most important observations (in mathematical terminology, the *statistics* of interest) are the total number $V(N)$ of different types in a sample of N tokens (also called the *vocabulary size*) and the number $V_m(N)$ of types that occur exactly m times

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 29–32,
Prague, June 2007.

in the sample. The set of values $V_m(N)$ for all frequency *ranks* $m = 1, 2, 3, \ldots$ is called a *frequency spectrum* and constitutes a sufficient statistic for the purpose of LNRE modeling.

A LNRE model M is a population model that specifies a certain distribution for the type probabilities in the population. This distribution can be linked to the observable values $V(N)$ and $V_m(N)$ by the standard assumption that the observed data are a *random sample* of size N from this population. It is most convenient mathematically to formulate a LNRE model in terms of a *type density function* $g(\pi)$, defined over the range of possible type probabilities $0 < \pi < 1$, such that $\int_a^b g(\pi)\, d\pi$ is the number of types with occurrence probabilities in the range $a \leq \pi \leq b$.[1] From the type density function, expected values $\mathrm{E}\big[V(N)\big]$ and $\mathrm{E}\big[V_m(N)\big]$ can be calculated with relative ease (Baayen, 2001), especially for the most widely-used LNRE models, which are based on Zipf's law and stipulate a power law function for $g(\pi)$. These models are known as GIGP (Sichel, 1975), ZM and fZM (Evert, 2004). For example, the type density of the ZM and fZM models is given by

$$g(\pi) := \begin{cases} C \cdot \pi^{-\alpha-1} & A \leq \pi \leq B \\ 0 & \text{otherwise} \end{cases}$$

with parameters $0 < \alpha < 1$ and $0 \leq A < B$. Baayen (2001) also presents approximate equations for the variances $\mathrm{Var}\big[V(N)\big]$ and $\mathrm{Var}\big[V_m(N)\big]$. In addition to such predictions for random samples, the type density $g(\pi)$ can also be used as a Bayesian prior, where it is especially useful for probability estimation from low-frequency data.

Baayen (2001) suggests a number of models that calculate the expected frequency spectrum directly without an underlying population model. While these models can sometimes be fitted very well to an observed frequency spectrum, they do not interpret the corpus data as a random sample from a population and hence do not allow for generalizations. They also cannot be used as a prior distribution for Bayesian inference. For these reasons, we do not see them as *proper* LNRE models and do not consider them useful for practical application.

3 Requirements and related software

As pointed out in the previous section, most applications of LNRE models rely on equations for the expected values and variances of $V(N)$ and $V_m(N)$ in a sample of arbitrary size N. The required basic operations are: (i) *parameter estimation*, where the parameters of a LNRE model M are determined from a training sample of size N_0 by comparing the expected frequency spectrum $\mathrm{E}\big[V_m(N_0)\big]$ with the observed spectrum $V_m(N_0)$; (ii) *goodness-of-fit* evaluation based on the covariance matrix of V and V_m; (iii) *interpolation* and *extrapolation* of vocabulary growth, using the expectations $\mathrm{E}\big[V(N)\big]$; and (iv) *prediction* of the expected frequency spectrum for arbitrary sample size N. In addition, Bayesian inference requires access to the type density $g(\pi)$ and distribution function $G(a) = \int_a^1 g(\pi)\, d\pi$, while random sampling from the population described by a LNRE model M is a prerequisite for Monte Carlo methods and simulation experiments.

Up to now, the only publicly available implementation of LNRE models has been the *lexstats* toolkit of Baayen (2001), which offers a wide range of models including advanced partition-adjusted versions and mixture models. While the toolkit supports the basic operations (i)–(iv) above, it does not give access to distribution functions or random samples (from the model distribution). It has not found widespread use among (computational) linguists, which we attribute to a number of limitations of the software: *lexstats* is a collection of command-line programs that can only be mastered with expert knowledge; an ad-hoc Tk-based graphical user interfaces simplifies basic operations, but is fully supported on the Linux platform only; the GUI also has only minimal functionality for visualization and data analysis; it has restrictive input options (making its use with languages other than English very cumbersome) and works reliably only for rather small data sets, well below the sizes now routinely encountered in linguistic research (cf. the problems reported in Evert and Baroni 2006); the standard parameter estimation methods are not very robust without extensive manual intervention, so *lexstats* cannot be used

[1] Since type probabilities are necessarily discrete, such a type density function can only give an approximation to the true distribution. However, the approximation is usually excellent for the low-probability types that are the center of interest for most applications of LNRE models.

as an off-the-shelf solution; and nearly all programs in the suite require interactive input, making it difficult to automate LNRE analyses.

4 Implementation

First and foremost, *zipfR* was conceived and developed to overcome the limitations of the *lexstats* toolkit. We implemented *zipfR* as an add-on library for the popular statistical computing environment R (R Development Core Team, 2003). It can easily be installed (from the CRAN archive) and used off-the-shelf for standard LNRE modeling applications. It fully supports the basic operations (i)–(iv), calculation of distribution functions and random sampling, as discussed in the previous section. We have taken great care to offer robust parameter estimation, while allowing advanced users full control over the estimation procedure by selecting from a wide range of optimization techniques and cost functions. In addition, a broad range of data manipulation techniques for word frequency data are provided. The integration of *zipfR* within the R environment makes the full power of R available for visualization and further statistical analyses.

For the reasons outlined above, our software package only implements proper LNRE models. Currently, the GIGP, ZM and fZM models are supported. We decided not to implement another LNRE model available in *lexstats*, the lognormal model, because of its numerical instability and poor performance in previous evaluation studies (Evert and Baroni, 2006).

More information about *zipfR* can be found on its homepage at *http://purl.org/stefan.evert/zipfR/*.

5 A sample session

In this section, we use a typical application example to give a brief overview of the basic functionality of the *zipfR* toolkit. *zipfR* accepts a variety of input formats, the most common ones being type frequency lists (which, in the simplest case, can be newline-delimited lists of frequency values) and tokenized (sub-)corpora (one word per line). Thus, as long as users can extract frequency data or at least tokenize the corpus of interest with other tools, they can perform all further analysis with *zipfR*.

Suppose that we want to compare the relative productivity of the Italian prefix *ri-* with that of the rarer prefix *ultra-* (roughly equivalent to English *re-* and *ultra-*, respectively), and that we have frequency lists of the word types containing the two prefixes.[2] In our R session, we import the data, create frequency spectra for the two classes, and we plot the spectra to look at their frequency distribution (the output graph is shown in the left panel of Figure 1):

```
ItaRi.tfl <- read.tfl("ri.txt")
ItaUltra.tfl <- read.tfl("ultra.txt")
ItaRi.spc <- tfl2spc(ItaRi.tfl)
ItaUltra.spc <- tfl2spc(ItaUltra.tfl)
> plot(ItaRi.spc,ItaUltra.spc,
+ legend=c("ri-","ultra-"))
```

We can then look at summary information about the distributions:

```
> summary(ItaRi.spc)
zipfR object for frequency spectrum
Sample size:     N  = 1399898
Vocabulary size: V  = 1098
Class sizes:     Vm = 346 105 74 43 ...
> summary(ItaUltra.spc)
zipfR object for frequency spectrum
Sample size:     N  = 3467
Vocabulary size: V  = 523
Class sizes:     Vm = 333 68 37 15 ...
```

We see that the *ultra-* sample is much smaller than the *ri-* sample, making a direct comparison of their vocabulary sizes problematic. Thus, we will use the fZM model (Evert, 2004) to estimate the parameters of the *ultra-* population (notice that the summary of an estimated model includes the parameters of the relevant distribution as well as goodness-of-fit information):

```
> ItaUltra.fzm <- lnre("fzm",ItaUltra.spc)
> summary(ItaUltra.fzm)
finite Zipf-Mandelbrot LNRE model.
Parameters:
   Shape:          alpha = 0.6625218
   Lower cutoff:       A = 1.152626e-06
   Upper cutoff:       B = 0.1368204
 [ Normalization:      C = 0.673407 ]
Population size: S = 8732.724
...
Goodness-of-fit (multivariate chi-squared):
         X2 df           p
   19.66858  5 0.001441900
```

Now, we can use the model to predict the frequency distribution of *ultra-* types at arbitrary sample sizes, including the size of our *ri-* sample. This allows us to compare the productivity of the two prefixes by using Baayen's $\mathscr{P}$, obtained by dividing the

[2] The data used for illustration are taken from an Italian newspaper corpus and are distributed with the toolkit.

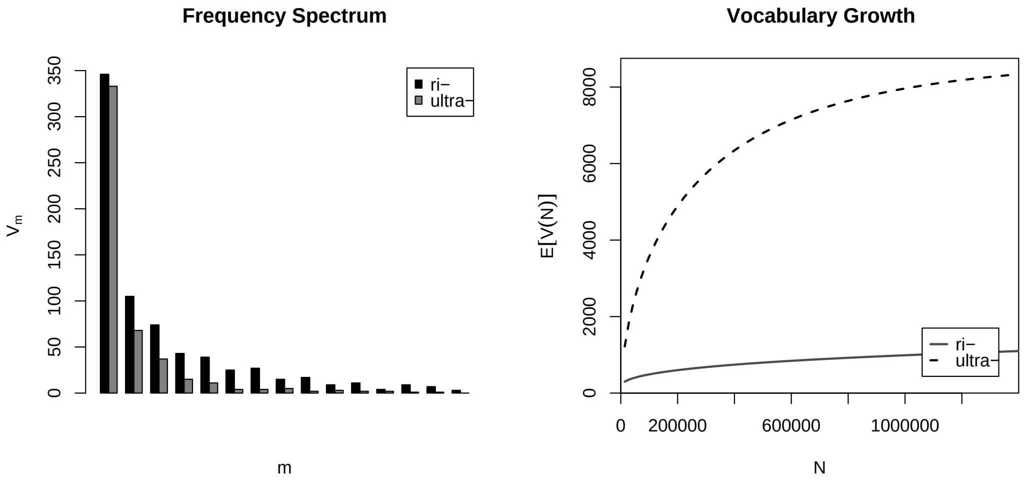


Figure 1: Left: Comparison of the observed *ri-* and *ultra-* frequency spectra. Right: Interpolated *ri-* vs. extrapolated *ultra-* vocabulary growth curves.

number of hapax legomena by the overall sample size (Baayen, 1992):

```
> ItaUltra.ext.spc<-lnre.spc(ItaUltra.fzm,
+ N(ItaRi.spc))
> Vm(ItaUltra.ext.spc,1)/N(ItaRi.spc)
[1] 0.0006349639
> Vm(ItaRi.spc,1)/N(ItaRi.spc)
[1] 0.0002471609
```

The rarer *ultra-* prefix appears to be more productive than the more frequent *ri-*. This is confirmed by a visual comparison of *vocabulary growth curves*, that report changes in vocabulary size as sample size increases. For *ri-*, we generate the growth curve by *binomial interpolation* from the observed spectrum, whereas for *ultra-* we extrapolate using the estimated LNRE model (Baayen 2001 discuss both techniques).

```
> sample.sizes <- floor(N(ItaRi.spc)/100)
+ *(1:100)
> ItaRi.vgc <- vgc.interp(ItaRi.spc,
+ sample.sizes)
> ItaUltra.vgc <- lnre.vgc(ItaUltra.fzm,
+ sample.sizes)
> plot(ItaRi.vgc,ItaUltra.vgc,
+ legend=c("ri-","ultra-"))
```

The plot (right panel of Figure 1) confirms the higher (potential) type richness of *ultra-*, a “fancier” prefix that is rarely used, but, when it does get used, is employed very productively (see discussion of similar prefixes in Gaeta and Ricca 2003).

References

Baayen, Harald. 1992. Quantitative aspects of morphological productivity. *Yearbook of Morphology 1991*, 109–150.

Baayen, Harald. 2001. *Word frequency distributions*. Dordrecht: Kluwer.

Evert, Stefan. 2004. A simple LNRE model for random character sequences. *Proceedings of JADT 2004*, 411–422.

Evert, Stefan and Marco Baroni. 2006. Testing the extrapolation quality of word frequency models. *Proceedings of Corpus Linguistics 2005*.

Gaeta, Livio and Davide Ricca. 2003. Italian prefixes and productivity: a quantitative approach. *Acta Linguistica Hungarica*, **50** 89–108.

Good, I. J. (1953). The population frequencies of species and the estimation of population parameters. *Biometrika*, **40**(3/4), 237–264.

R Development Core Team (2003). *R: A language and environment for statistical computing*. R Foundation for Statistical Computing, Vienna, Austria. ISBN 3-900051-00-3. See also `http://www.r-project.org/`.

Sichel, H. S. (1975). On a distribution law for word frequencies. *Journal of the American Statistical Association*, **70**, 542–547.

Zipf, George K. 1949. *Human behavior and the principle of least effort*. Cambridge (MA): Addison-Wesley.

Linguistically Motivated Large-Scale NLP with C&C and Boxer

James R. Curran
School of Information Technologies
University of Sydney
NSW 2006, Australia
james@it.usyd.edu.au

Stephen Clark
Computing Laboratory
Oxford University
Wolfson Building, Parks Road
Oxford, OX1 3QD, UK
stephen.clark@comlab.ox.ac.uk

Johan Bos
Dipartimento di Informatica
Università di Roma "La Sapienza"
via Salaria 113
00198 Roma, Italy
bos@di.uniroma1.it

1 Introduction

The statistical modelling of language, together with advances in wide-coverage grammar development, have led to high levels of robustness and efficiency in NLP systems and made linguistically motivated large-scale language processing a possibility (Matsuzaki et al., 2007; Kaplan et al., 2004). This paper describes an NLP system which is based on syntactic and semantic formalisms from theoretical linguistics, and which we have used to analyse the entire Gigaword corpus (1 billion words) in less than 5 days using only 18 processors. This combination of detail and speed of analysis represents a breakthrough in NLP technology.

The system is built around a wide-coverage Combinatory Categorial Grammar (CCG) parser (Clark and Curran, 2004b). The parser not only recovers the local dependencies output by treebank parsers such as Collins (2003), but also the long-range depdendencies inherent in constructions such as extraction and coordination. CCG is a lexicalized grammar formalism, so that each word in a sentence is assigned an elementary syntactic structure, in CCG's case a lexical category expressing subcategorisation information. Statistical tagging techniques can assign lexical categories with high accuracy and low ambiguity (Curran et al., 2006). The combination of finite-state supertagging and highly engineered C++ leads to a parser which can analyse up to 30 sentences per second on standard hardware (Clark and Curran, 2004a).

The C&C tools also contain a number of Maximum Entropy taggers, including the CCG supertagger, a POS tagger (Curran and Clark, 2003a), chunker, and named entity recogniser (Curran and Clark, 2003b). The taggers are highly efficient, with processing speeds of over 100,000 words per second.

Finally, the various components, including the morphological analyser morpha (Minnen et al., 2001), are combined into a single program. The output from this program — a CCG derivation, POS tags, lemmas, and named entity tags — is used by the module Boxer (Bos, 2005) to produce interpretable structure in the form of Discourse Representation Structures (DRSs).

2 The CCG Parser

The grammar used by the parser is extracted from CCGbank, a CCG version of the Penn Treebank (Hockenmaier, 2003). The grammar consists of 425 lexical categories, expressing subcategorisation information, plus a small number of combinatory rules which combine the categories (Steedman, 2000). A Maximum Entropy supertagger first assigns lexical categories to the words in a sentence (Curran et al., 2006), which are then combined by the parser using the combinatory rules and the CKY algorithm.

Clark and Curran (2004b) describes log-linear parsing models for CCG. The features in the models are defined over local parts of CCG derivations and include word-word dependencies. A disadvantage of the log-linear models is that they require cluster computing resources for practical training (Clark and Curran, 2004b). We have also investigated perceptron training for the parser (Clark and Curran, 2007b), obtaining comparable accuracy scores and similar training times (a few hours) compared with the log-linear models. The significant advantage of

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 33–36,
Prague, June 2007.

the perceptron training is that it only requires a single processor. The training is online, updating the model parameters one sentence at a time, and it converges in a few passes over the CCGbank data.

A packed chart representation allows efficient decoding, with the same algorithm — the Viterbi algorithm — finding the highest scoring derivation for the log-linear and perceptron models.

2.1 The Supertagger

The supertagger uses Maximum Entropy tagging techniques (Section 3) to assign a set of lexical categories to each word (Curran et al., 2006). Supertagging has been especially successful for CCG: Clark and Curran (2004a) demonstrates the considerable increases in speed that can be obtained through use of a supertagger. The supertagger interacts with the parser in an adaptive fashion: initially it assigns a small number of categories, on average, to each word in the sentence, and the parser attempts to create a spanning analysis. If this is not possible, the supertagger assigns more categories, and this process continues until a spanning analysis is found.

2.2 Parser Output

The parser produces various types of output. Figure 1 shows the dependency output for the example sentence *But Mr. Barnum called that a worst-case scenario.* The CCG dependencies are defined in terms of the arguments within lexical categories; for example, $\langle (S[dcl]\backslash NP_1)/NP_2, 2\rangle$ represents the direct object of a transitive verb. The parser also outputs grammatical relations (GRs) consistent with Briscoe et al. (2006). The GRs are derived through a manually created mapping from the CCG dependencies, together with a python post-processing script which attempts to remove any differences between the two annotation schemes (for example the way in which coordination is analysed).

The parser has been evaluated on the predicate-argument dependencies in CCGbank, obtaining labelled precision and recall scores of 84.8% and 84.5% on Section 23. We have also evaluated the parser on DepBank, using the Grammatical Relations output. The parser scores 82.4% labelled precision and 81.2% labelled recall overall. Clark and Curran (2007a) gives precison and recall scores broken down by relation type and also compares the performance of the CCG parser with the RASP parser (Briscoe et al., 2006).

```
Mr._2 N/N_1 1 Barnum_3
called_4 ((S[dcl]\NP_1)/NP_2)/NP_3 3 that_5
worst-case_7 N/N_1 1 scenario_8
a_6 NP[nb]/N_1 1 scenario_8
called_4 ((S[dcl]\NP_1)/NP_2)/NP_3 2 scenario_8
called_4 ((S[dcl]\NP_1)/NP_2)/NP_3 1 Barnum_3
But_1 S[X]/S[X]_1 1 called_4

(ncmod _ Barnum_3 Mr._2)
(obj2 called_4 that_5)
(ncmod _ scenario_8 worst-case_7)
(det scenario_8 a_6)
(dobj called_4 scenario_8)
(ncsubj called_4 Barnum_3 _)
(conj _ called_4 But_1)
```

Figure 1: Dependency output in the form of CCG dependencies and grammatical relations

3 Maximum Entropy Taggers

The taggers are based on Maximum Entropy tagging methods (Ratnaparkhi, 1996), and can all be trained on new annotated data, using either GIS or BFGS training code.

The POS tagger uses the standard set of grammatical categories from the Penn Treebank and, as well as being highly efficient, also has state-of-the-art accuracy on unseen newspaper text: over 97% per-word accuracy on Section 23 of the Penn Treebank (Curran and Clark, 2003a). The chunker recognises the standard set of grammatical “chunks”: NP, VP, PP, ADJP, ADVP, and so on. It has been trained on the CoNLL shared task data.

The named entity recogniser recognises the standard set of named entities in text: person, location, organisation, date, time, monetary amount. It has been trained on the MUC data. The named entity recogniser contains many more features than the other taggers; Curran and Clark (2003b) describes the feature set.

Each tagger can be run as a “multi-tagger”, potentially assigning more than one tag to a word. The multi-tagger uses the forward-backward algorithm to calculate a distribution over tags for each word in the sentence, and a parameter determines how many tags are assigned to each word.

4 Boxer

Boxer is a separate component which takes a CCG derivation output by the C&C parser and generates a semantic representation. Boxer implements a first-order fragment of Discourse Representation Theory, DRT (Kamp and Reyle, 1993), and is capable of generating the box-like structures of DRT known as Discourse Representation Structures (DRSs). DRT is a formal semantic theory backed up with a model theory, and it demonstrates a large coverage of linguistic phenomena. Boxer follows the formal theory closely, introducing discourse referents for noun phrases and events in the domain of a DRS, and their properties in the conditions of a DRS.

One deviation with the standard theory is the adoption of a Neo-Davidsonian analysis of events and roles. Boxer also implements Van der Sandt's theory of presupposition projection treating proper names and defininite descriptions as anaphoric expressions, by binding them to appropriate previously introduced discourse referents, or accommodating on a suitable level of discourse representation.

4.1 Discourse Representation Structures

DRSs are recursive data structures — each DRS comprises a domain (a set of discourse referents) and a set of conditions (possibly introducing new DRSs). DRS-conditions are either basic or complex. The basic DRS-conditions supported by Boxer are: equality, stating that two discourse referents refer to the same entity; one-place relations, expressing properties of discourse referents; two place relations, expressing binary relations between discourse referents; and names and time expressions. Complex DRS-conditions are: negation of a DRS; disjunction of two DRSs; implication (one DRS implying another); and propositional, relating a discourse referent to a DRS.

Nouns, verbs, adjectives and adverbs introduce one-place relations, whose meaning is represented by the corresponding lemma. Verb roles and prepositions introduce two-place relations.

4.2 Input and Output

The input for Boxer is a list of CCG derivations decorated with named entities, POS tags, and lemmas for nouns and verbs. By default, each CCG derivation produces one DRS. However, it is possible for one DRS to span several CCG derivations; this enables Boxer to deal with cross-sentential phenomena such as pronouns and presupposition.

Boxer provides various output formats. The default output is a DRS in Prolog format, with discourse referents represented as Prolog variables. Other output options include: a flat structure, in which the recursive structure of a DRS is unfolded by labelling each DRS and DRS-condition; an XML format; and an easy-to-read box-like structure as found in textbooks and articles on DRT. Figure 2 shows the easy-to-read output for the sentence *But Mr. Barnum called that a worst-case scenario.*

```
 ______________________
| x0 x1 x2 x3          |
|______________________|
| named(x0,barnum,per) |
| named(x0,mr,ttl)     |
| thing(x1)            |
| worst-case(x2)       |
| scenario(x2)         |
| call(x3)             |
| but(x3)              |
| event(x3)            |
| agent(x3,x0)         |
| patient(x3,x1)       |
| theme(x3,x2)         |
|______________________|
```

Figure 2: Easy-to-read output format of Boxer

The semantic representations can also be output as first-order formulas. This is achieved using the standard translation from DRS to first-order logic (Kamp and Reyle, 1993), and allows the output to be pipelined into off-the-shelf theorem provers or model builders for first-order logic, to perform consistency or informativeness checking (Blackburn and Bos, 2005).

5 Usage of the Tools

The taggers (and therefore the parser) can accept many different input formats and produce many different output formats. These are described using a "little language" similar to C printf format strings. For example, the input format %w|%p \n indicates that the program expects word (%w) and POS tag (%p) pairs as input, where the words and POS tags are separated by pipe characters, and each word-POS tag pair is separated by a single space, and whole sentences are separated by newlines (\n). Another feature of the input/output is that other fields can be read in which are not used in the tagging process, and also form part of the output.

The C&C tools use a configuration management system which allows the user to override all of the default parameters for training and running the taggers and parser. All of the tools can be used as standalone components. Alternatively, a pipeline of the tools is provided which supports two modes: local file reading/writing or SOAP server mode.

6 Applications

We have developed an open-domain QA system built around the C&C tools and Boxer (Ahn et al., 2005). The parser is well suited to analysing large amounts of text containing a potential answer, because of its efficiency. The grammar is also well suited to analysing questions, because of CCG's treatment of long-range dependencies. However, since the CCG parser is based on the Penn Treebank, which contains few examples of questions, the parser trained on CCGbank is a poor analyser of questions. Clark et al. (2004) describes a porting method we have developed which exploits the lexicalized nature of CCG by relying on rapid manual annotation at the lexical category level. We have successfully applied this method to questions.

The robustness and efficiency of the parser; its ability to analyses questions; and the detailed output provided by Boxer make it ideal for large-scale open-domain QA.

7 Conclusion

Linguistically motivated NLP can now be used for large-scale language processing applications. The C&C tools plus Boxer are freely available for research use and can be downloaded from http://svn.ask.it.usyd.edu.au/trac/candc/wiki.

Acknowledgements

James Curran was funded under ARC Discovery grants DP0453131 and DP0665973. Johan Bos is supported by a "Rientro dei Cervelli" grant (Italian Ministry for Research).

References

Kisuh Ahn, Johan Bos, James R. Curran, Dave Kor, Malvina Nissim, and Bonnie Webber. 2005. Question answering with QED at TREC-2005. In *Proceedings of TREC-2005*.

Patrick Blackburn and Johan Bos. 2005. *Representation and Inference for Natural Language. A First Course in Computational Semantics*. CSLI.

Johan Bos. 2005. Towards wide-coverage semantic interpretation. In *Proceedings of IWCS-6*, pages 42–53, Tilburg, The Netherlands.

Ted Briscoe, John Carroll, and Rebecca Watson. 2006. The second release of the RASP system. In *Proceedings of the Interactive Demo Session of COLING/ACL-06*, Sydney.

Stephen Clark and James R. Curran. 2004a. The importance of supertagging for wide-coverage CCG parsing. In *Proceedings of COLING-04*, pages 282–288, Geneva, Switzerland.

Stephen Clark and James R. Curran. 2004b. Parsing the WSJ using CCG and log-linear models. In *Proceedings of ACL-04*, pages 104–111, Barcelona, Spain.

Stephen Clark and James R. Curran. 2007a. Formalism-independent parser evaluation with CCG and DepBank. In *Proceedings of the 45th Annual Meeting of the ACL*, Prague, Czech Republic.

Stephen Clark and James R. Curran. 2007b. Perceptron training for a wide-coverage lexicalized-grammar parser. In *Proceedings of the ACL Workshop on Deep Linguistic Processing*, Prague, Czech Republic.

Stephen Clark, Mark Steedman, and James R. Curran. 2004. Object-extraction and question-parsing using CCG. In *Proceedings of the EMNLP Conference*, pages 111–118, Barcelona, Spain.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *Computational Linguistics*, 29(4):589–637.

James R. Curran and Stephen Clark. 2003a. Investigating GIS and smoothing for maximum entropy taggers. In *Proceedings of the 10th Meeting of the EACL*, pages 91–98, Budapest, Hungary.

James R. Curran and Stephen Clark. 2003b. Language independent NER using a maximum entropy tagger. In *Proceedings of CoNLL-03*, pages 164–167, Edmonton, Canada.

James R. Curran, Stephen Clark, and David Vadas. 2006. Multi-tagging for lexicalized-grammar parsing. In *Proceedings of COLING/ACL-06*, pages 697–704, Sydney.

Julia Hockenmaier. 2003. *Data and Models for Statistical Parsing with Combinatory Categorial Grammar*. Ph.D. thesis, University of Edinburgh.

H. Kamp and U. Reyle. 1993. *From Discourse to Logic; An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and DRT*. Kluwer, Dordrecht.

Ron Kaplan, Stefan Riezler, Tracy H. King, John T. Maxwell III, Alexander Vasserman, and Richard Crouch. 2004. Speed and accuracy in shallow and deep stochastic parsing. In *Proceedings of HLT and the 4th Meeting of NAACL*, Boston, MA.

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2007. Efficient HPSG parsing with supertagging and CFG-filtering. In *Proceedings of IJCAI-07*, Hyderabad, India.

Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of English. *Natural Language Engineering*, 7(3):207–223.

Adwait Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Proceedings of the EMNLP Conference*, pages 133–142, Philadelphia, PA.

Mark Steedman. 2000. *The Syntactic Process*. The MIT Press, Cambridge, MA.

Don't worry about metaphor: affect extraction for conversational agents

Catherine Smith, Tim Rumbell, John Barnden, Bob Hendley, Mark Lee & Alan Wallington
School of Computer Science, University of Birmingham
Birmingham B15 2TT, UK
J.A.Barnden@cs.bham.ac.uk

Abstract

We demonstrate one aspect of an affect-extraction system for use in intelligent conversational agents. This aspect performs a degree of affective interpretation of some types of metaphorical utterance.

1 Introduction

Our demonstration is of one aspect of a system for extracting affective information from individual utterances, for use in text-based intelligent conversational agents (ICAs). Affect includes emotions/moods (such as embarrassment, hostility) and evaluations (of goodness, importance, etc.). Our own particular ICA [Zhang *et al.* 2006] is for use in an e-drama system, where human users behave as actors engaged in unscripted role-play. Actors type in utterances for the on-screen characters they control to utter (via speech bubbles). Our ICA is another actor, controlling a bit-part character. Through extracting affect from other characters' utterances it makes responses that can help keep the conversation flowing. The same algorithms are also used for influencing the characters' gesturing (when a 3D animation mode is used).

The system aspect demonstrated handles one important way in which affect is expressed in most discourse genres: namely metaphor. Only a relatively small amount of work has been done on computational processing of metaphorical meaning, for any purpose, let alone in ICA research. Major work apart from ours on metaphorical-meaning computation includes (Fass, 1997; Hobbs, 1990; Martin, 1990; Mason, 2004; Narayanan, 1999; Veale, 1998). The e-drama genre exhibits a variety of types of metaphor, with a significant degree of linguistic open-endedness. Also, note that our overarching research aim is to study metaphor as such, not just how it arises in e-drama. This increases our need for systematic, open-ended methods.

2 Metaphor and Affect

Conveying affect is one important role for metaphor, and metaphor is one important way of conveying affect. Emotional states and behavior often themselves described metaphorically (Kövecses, 2000; Fussell & Moss, 1998), as in 'He was boiling inside' [feelings of anger]. But another important phenomenon is describing something X using metaphorical source terms that are subject to that affect, as in 'My son's room [= X] is a *bomb site*' or '*smelly* attitude' (an e-drama transcript example). Such carry-over of affect in metaphor is well-recognized, e.g. in the political domain (Musolff, 2004). Our transcript analyses indicate that this type of affect-laden metaphor is a significant issue in e-drama: at a conservative estimate, in recent user studies in secondary schools at least one in every 16 speech-turns has contained such metaphor (each turn is $\leq$100 characters, and rarely more than one sentence; 33K words across all transcripts).

There are other specific, theoretically interesting metaphorical phenomena arising in e-drama that are important also for discourse in general, and plausibly could be handled reasonably successfully in an ICA using current techniques. Some are:
1) Casting someone as an animal. This often conveys affect, from insultingly negative to affectionately positive. Terms for young animals ('piglet', 'wolf cub', etc.) are often used affectionately, even

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 37–40,
Prague, June 2007. ©2007 Association for Computational Linguistics

when the adult form is negative. Animal words can have a conventional metaphorical sense, often with specific affect, but in non-conventional cases a system may still be able to discern a particular affective connotation; and even if it cannot, it can still plausibly infer that *some* affect is expressed, of unknown polarity (positivity/negativity).
2) Rather similarly, casting someone as a monster or as a mythical or supernatural being, using words such as 'monster', 'dragon,' 'angel,' 'devil.'
3) Casting someone as a special type of human, using words such as 'baby' (to an adult), 'freak,' 'girl' (to a boy), 'lunatic.'
4) Metaphorical use of size adjectives (cf. Sharoff, 2006). Particularly, using 'a little X' to convey affective qualities of X such as unimportance and contemptibility, but sometimes affection towards X, and 'big X' to convey importance of X ('big event') or intensity of X-ness ('big bully')—and X can itself be metaphorical ('baby', 'ape').
Currently, our system partially addresses (1), (2) and (4).

3 Metaphor Recognition & Analysis

3.1 The Recognition Component

The basis here is a subset of a list of metaphoricity signals we have compiled [`http://www.cs.bham.ac.uk/˜jab/ATT-Meta/metaphoricity-signals.html`], by modifying and expanding a list from Goatly (1997). The signals include specific syntactic structures, phraseological items and morphological elements. We currently focus on two special syntactic structures, *X is/are Y* and *You/you Y*, and some lexical strings such as '[looks] like', 'a bit of a' and 'such a'. The signals are merely uncertain, heuristic indicators. For instance, in the transcripts mentioned in section 2, we judged *X is/are Y* as actually indicating the presence of metaphor in 38% of cases (18 out of 47). Other success rates are: *you Y* – 61% (22 out of 36); *like* (including *looks like*)– 81% (35 out of 43).

In order to detect signals we use the Grammatical Relations (GR) output from the RASP robust parser [Briscoe *et al.*, 2006] This output shows typed word-pair dependencies between the words in the utterance. E.g., the GR output for 'You are a pig' is:

```
|ncsubj| |be+_vbr| |you_ppy| |_|
|xcomp| _ |be+_vbr| |pig_nn1|
|det| |pig_nn1| |a_at1|
```

For an utterance of the type *X is/are Y* the GRs will always give a subject relation (`ncsubj`) between X and the verb 'to be', as well as a complement relation (`xcomp`) between the verb and the noun Y. The structure is detected by finding these relations. As for *you Y*, Rasp also typically delivers an easily analysable structure, but unfortunately the POS tagger in Rasp seems to favour tagging Y as a verb—e.g., 'cow' in 'You cow'. In such a case, our system looks the word up in a list of tagged words that forms part of the RASP tagger. If the verb can be tagged as a noun, the tag is changed, and the metaphoricity signal is deemed detected. Once a signal is detected, the word(s) in relevant positions (e.g. the Y position) position are pulled out to be analysed. This approach has the advantage that whether or not the noun in, say, the Y position has adjectival modifiers the GR between the verb and Y is the same, so the detection tolerates a large amount of variation. Any such modifiers are found in modifying relations and are available for use in the Analysis Component.

3.2 The Analysis Component

We confine attention here to *X–is/are–Y* and *You–Y* cases. The analysis element of the processing takes the X noun (if any) and Y noun and uses WordNet 2.0 to analyse them. First, we try to determine whether X refers to a person (the only case the system currently deals with), partly by using a specified list of proper names of characters in the drama and partly by WordNet processing. If so, then the Y and remaining elements are analysed using WordNet's taxonomy. This allows us to see if the Y noun in one of its senses is a hyponym of animals or supernatural beings. If this is established, the system sees if another of the senses of the word is a hyponym of the person synset, as many metaphors are already given as senses in WordNet. If different senses of the given word are hyponyms of both animal and person, other categories in the tree between the noun and the person synset may provide information about the evaluative content of the metaphor. For example the word 'cow' in its metaphorical usage has the 'unpleasant person' synset as a lower hypernym, which heuristically suggests that, when the word is used in a metaphor, it will be negative about the target.

There is a further complication. Baby animal names can often be used to give a statement a more affectionate quality. Some baby animal names such as 'piglet' do not have a metaphorical sense in Word-

Net. In these cases, we check the word's gloss to see if it is a young animal and what kind of animal it is. We then process the adult animal name to seek a metaphorical meaning but add the quality of affection to the result. A higher degree of confidence is attached to the quality of affection than is attached to the positive/negative result, if any, obtained from the adult name. Other baby animal names such as 'lamb' do have a metaphorical sense in WordNet independently of the adult animal, and are therefore evaluated as in the previous paragraph. They are also flagged as potentially expressing affection but with a lesser degree of confidence than that gained from the metaphorical processing of the word. However, the youth of an animal is not always encoded in a single word: e.g., 'cub' may be accompanied by specification of an animal type, as in 'wolf cub'. An extension to our processing would be required to handle this and also cases like 'young wolf' or 'baby wolf'.

If any adjectival modifiers of the Y noun were recognized the analyser then goes on to evaluate their contribution to the metaphor's affect. If the analyser finds that 'big' is one of the modifying adjectives of the noun it has analysed the metaphor is marked as being more emphatic. If 'little' is found the following is done. If the metaphor has been tagged as negative and no degree of affection has been added (from a baby animal name, currently) then 'little' is taken to be expressing contempt. If the metaphor has been tagged as positive OR a degree of affection has been added then 'little' is taken to be expressing affection.

4 Examples of Course of Processing

'You piglet':

(1) Detector recognises the *you Y* signal with Y = 'piglet'.

(2) Analyser finds that 'piglet' is a hyponym of 'animal'.

(3) 'Piglet' does not have 'person' as a WordNet hypernym so analyser retrieves the WordNet gloss.

(4) It finds 'young' in the gloss ('a young pig') and retrieves all of the following words (just 'pig' – the analysis process is would otherwise be repeated for each of the words captured from the gloss), and finds that 'pig' by itself has negative metaphorical affect.

(5) The input is labelled as an animal metaphor which is negative but affectionate, with the affection having higher confidence than the negativity.

'Lisa is an angel':

(1) Detector recognises the *X is Y* signal with Y = 'angel', after checking that Lisa is a person.

(2) Analyser finds that 'angel' is a hyponym of 'supernatural being'.

(3) It finds that in another sense 'angel' is a hyponym of 'person'.

(4) It finds that the tree including the 'person' synset also passes through 'good person,' expressing positive affect.

(5) Conclusion: positive supernatural-being metaphor.

Results from Some Other Examples:

"You cow", "they're such sheep": negative metaphor.

"You little rat": contemptuous metaphor.

"You little piggy": affectionate metaphor with a negative base.

"You're a lamb": affectionate metaphor.

"You are a monster": negative metaphor.

"She is such a big fat cow": negative metaphor, intensified by 'big' (currently 'fat' is not dealt with).

5 Concluding Remarks

The demonstrated processing capabilities make particular but nevertheless valuable contributions to metaphor processing and affect-detection for ICAs, in e-drama at least. Further work is ongoing on the four specific metaphorical phenomena in section 3 as well as on other phenomena, such as the variation of conventional metaphorical phraseology by synonym substitution and addition of modifiers, and metaphorical descriptions of emotions themselves.

As many extensions are ongoing or envisaged, it is premature to engage in large-scale evaluation. Also, there are basic problems facing evaluation. The language in the e-drama genre is full of misspellings, "texting" abbreviations, acronyms, grammatical errors, etc., so that fully automated evaluation of the metaphorical processing by itself is difficult; and application of the system to manually cleaned-up utterances is still dependent on Rasp extracting structure appropriately. Also, our own ultimate concerns are theoretical, to do with the nature of metaphor understanding. We are interested in covering the qualitative range of possibilities and complications, with no strong constraint from their frequency in real discourse. Thus, statistical evaluation on corpora is not particularly relevant except for practical purposes.

However, some proto-evaluative comments that can be made about animal metaphors are as follows. The transcripts mentioned in section 2 (33K words total) contain metaphors with the following animal words: *rhino, bitch, dog, ape, cow, mole*, from 14 metaphorical utterances in all. Seven of the utterances are recognized by our system, and these involve *rhino, dog, ape, mole.* No WordNet-based metaphorical connotation is found for the *rhino* case. Negative affect is concluded for *bitch, dog* and *cow* cases, and affect of undetermined polarity is concluded for *ape* and *mole*.

The system is currently designed only to do relatively simple, specialized metaphorical processing. The system currently only deals with a small minority of our own list of metaphoricity signals (see section 3.1), and these signals are only present in a minority of cases of metaphor overall. It does not do either complex reasoning or analogical structure-matching as in our own ATT-Meta metaphor system (Barnden, 2006) or the cited approaches of Fass, Hobbs, Martin, Narayanan and Veale. However, we plan to eventually add simplified versions of ATT-Meta-style reasoning, and in particular to add the ATT-Meta *view-neutral mapping adjunct* feature to implement the default carry-over of affect (see section 2) and certain other information, as well as handling more signals.

Other work on metaphor has exploited WordNet (see, e.g., Veale, 2003, and panel on Figurative Language in WordNets and other Lexical Resources at GWC'04 (`http://www.fi.muni.cz/gwc2004/`. Such work uses WordNet in distinctly different ways from us and largely for different purposes. Our system is also distinctive in, for instance, interpreting the contribution of size adjectives.

Acknowledgments

The research has been aided by Sheila Glasbey and Li Zhang, and supported by ESRC/EPSRC/DTI Paccit LINK grant (ESRC RES-328-25-0009) and EPSRC grant EP/C538943/1.

References

John Barnden. 2006. Artificial Intelligence, Figurative Language and Cognitive Linguistics. In G. Kristiansen *et al.* (Eds), *Cognitive Linguistics: Current Applications and Future Perspectives,* 431–459. Berlin: Mouton de Gruyter.

Ted Briscoe, John Carroll and Rebecca Watson. 2006. The Second Release of the RASP System. In *Procs. COLING/ACL 2006 Interactive Presentation Sessions*. Sydney, Australia.

Dan Fass. 1997. *Processing Metaphor and Metonymy.* Greenwich, Connecticut: Ablex.

Susan Fussell & Mallie Moss. 1998. Figurative Language in Emotional Communication. *Social and Cognitive Approaches to Interpersonal Communication*. Lawrence Erlbaum.

Andrew Goatly. 1997. *The Language of Metaphors.* Routledge, London.

Jerry Hobbs. 1990. *Literature and Cognition.* CSLI Lecture Notes, 21, Stanford University, 1990.

Zoltán Kövecses. 2000. *Metaphor and Emotion: Language, Culture and Body in Human Feeling*. Cambridge University Press, Cambridge.

James Martin. 1990. *A Computational Model of Metaphor Interpretation.* Academic Press.

Zachary Mason. 2004. CorMet: A computational, corpus-based conventional metaphor extraction system. *Computational Linguistics*, 30(1), 23–44.

Andreas Musolff. 2004. *Metaphor and political discourse: Analogical reasoning in debates about Europe.* Palgrave Macmillan.

Srini Narayanan. 1999. Moving right along: A computational model of metaphoric reasoning about events. *Procs. National Conference on Art. Int.*, 121–128.

Serge Sharoff. 2006. How to Handle Lexical Semantics in SFL: A Corpus Study of Purposes for Using Size Adjectives. *System and Corpus: Exploring Connections*. Equinox, London.

Tony Veale. 1998. 'Just in Time' analogical mapping, an iterative-deepening approach to structure-mapping. In *Procs. 13th European Conference on Art. Intell.*

Tony Veale. 2003. Dynamic Type Creation in Metaphor Interpretation and Analogical Reasoning: A Case-Study with WordNet. In *Procs. International Conference on Conceptual Structures* (Dresden).

Li Zhang, John Barnden, Bob Hendley & Alan Wallington. 2006. Exploitation in Affect Detection in Improvisational E-Drama. In Procs. 6th Int. Conference on Intelligent Virtual Agents: *Lecture Notes in Computer Science, 4133*, 68–79. Springer.

An efficient algorithm for building a distributional thesaurus (and other Sketch Engine developments)

Pavel Rychlý
Masaryk University
Brno, Czech Republic
pary@fi.muni.cz

Adam Kilgarriff
Lexical Computing Ltd
Brighton, UK
adam@lexmasterclass.com

Abstract

Gorman and Curran (2006) argue that thesaurus generation for billion+-word corpora is problematic as the full computation takes many days. We present an algorithm with which the computation takes under two hours. We have created, and made publicly available, thesauruses based on large corpora for (at time of writing) seven major world languages. The development is implemented in the Sketch Engine (Kilgarriff et al., 2004).

Another innovative development in the same tool is the presentation of the grammatical behaviour of a word against the background of how all other words of the same word class behave. Thus, the English noun *constraint* occurs 75% in the plural. Is this a salient lexical fact? To form a judgement, we need to know the distribution for all nouns. We use histograms to present the distribution in a way that is easy to grasp.

1 Thesaurus creation

Over the last ten years, interest has been growing in distributional thesauruses (hereafter simply 'thesauruses'). Following initial work by (Spärck Jones, 1964) and (Grefenstette, 1994), an early, online distributional thesaurus presented in (Lin, 1998) has been widely used and cited, and numerous authors since have explored thesaurus properties and parameters: see survey component of (Weeds and Weir, 2005).

A thesaurus is created by

- taking a corpus
- identifying contexts for each word
- identifying which words share contexts.

For each word, the words that share most contexts (according to some statistic which also takes account of their frequency) are its nearest neighbours.

Thesauruses generally improve in accuracy with corpus size. The larger the corpus, the more clearly the signal (of similar words) will be distinguished from the noise (of words that just happen to share a few contexts). Lin's was based on around 300M words and (Curran, 2004) used 2B (billion).

A direct approach to thesaurus computation looks at each word and compares it with each other word, checking all contexts to see if they are shared. Thus, complexity is $O(n^2m)$ where n in the number of types and m is the size of the context vector. The number of types increases with the corpus size, and (Ravichandran et al., 2005) propose heuristics for thesaurus building without undertaking the complete calculation. The line of reasoning is explored further by (Gorman and Curran, 2006), who argue that the complete calculation is not realistic given large corpora. They estimate that, given a 2B corpus and its 184,494-word vocabulary comprising all words occurring over five times, the full calculation will take nearly 300 days. With the vocabulary limited to the 75,800 words occuring over 100 times, the calculation took 18 days.

The naive algorithm has complexity $O(n^2m)$ but this is not the complexity of the problem. Most of

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 41–44,
Prague, June 2007.

the n^2 word pairs have nothing in common so there is no reason to check them. We proceed by working only with those word pairs that do have something in common. This allows us to create thesauruses from 1B corpora in under 2 hours.

1.1 Algorithm

We prepare the corpus by lemmatizing and then shallow parsing to identify grammatical relation instances with the form $\langle w_1, r, w' \rangle$, where r is a grammatical relation, w_1 and w' are words. We count the frequency of each triple and sort all $\langle w_1, r, w', score \rangle$ 4-tuples by 'contexts' where a context is a $\langle r, w' \rangle$ pair. Only 4-tuples with positive $score$ are included.

The algorithm then loops over each context (CONTEXTS is the set of all contexts):

```
for $\langle r, w' \rangle$ in CONTEXTS:
    WLIST = set of all $w$ where $\langle w, r, w' \rangle$ exists
    for $w_1$ in WLIST:
        for $w_2$ in WLIST:
            $sim(w_1, w_2) += f(frequencies)$[1]
```

The outer loop is linear in the number of contexts. The inner loop is quadratic in the number of words in WLIST, that is, the number of words sharing a particular context $\langle r, w' \rangle$. This list is usually small (less than 1000), so the quadratic complexity is manageable.

We use a heuristic at this point. If WLIST has more than 10,000 members, the context is skipped. Any such general context is very unlikely to make a substantial difference to the similarity score, since similarity scores are weighted according to how specific they are. The computational work avoided can be substantial.

The next issue is how to store the whole $sim(w_1, w_2)$ matrix. Most of the values are very small or zero. These values are not stored in the final thesaurus but they are needed during the computation. A strategy for this problem is to generate, sort and sum in sequential scan. That means that instead of incrementing the $sim(w_1, w_2)$ score as we go along, we produce $\langle w_1, w_2, x \rangle$ triples in a very long list, running, for a billion-word corpus, into hundreds of GB. For such huge data, a variant of TPMMS (Two Phase Multi-way Merge Sort) is used. First we fill the whole available memory with a part of the data, sort in memory (summing where we have multiple instances of the same $\langle w_1, w_2 \rangle$ as we proceed) and output the sorted stream. Then we merge sorted streams, again summing as we proceed.

Another technique we use is partitioning. The outer loop of the algorithm is fast and can be run several times with a limit on which words to process and output. For example, the first run processes only word pairs $\langle w_1, w_2 \rangle$ where the ID of w_1 is between 0 and 99, the next, where it is between 100 and 199, etc. In such limited runs there is a high probability that most of the summing is done in memory. We establish a good partitioning with a dry run in which a plan is computed such that all runs produce approximately the number of items which can be sorted and summed in memory.

1.2 Experiments

We experimented with the 100M-word BNC[2], 1B-word Oxford English Corpus[3] (OEC), and 1.9B-word Itwac (Baroni and Kilgarriff, 2006).

All experiments were carried out on a machine with AMD Opteron quad-processor. The machine has 32 GB of RAM but each process used only 1GB (and changing this limit produced no significant speedup). Data files were on a Promise disk array running Disk RAID5.

Parameters for the computation include:

- hits threshold MIN: only words entering into a number of triples greater than MIN will have thesaurus entries, or will be candidates for being in other words' thesaurus entries. (Note that words not passing this threshold can still be in contexts, so may contribute to the similarity of two other words: cf Daelemans et al.'s title (1999).)

- the number of words (WDS) above the threshold

[1] In this paper we do not discuss the nature of this function as it is does not impact on the complexity. It is explored extensively in (Curran, 2004; Weeds and Weir, 2005).

[2] http://www.natcorp.ox.ac.uk

[3] http://www.askoxford.com/oec/ We are grateful to Oxford University Press for permission to use the OEC.

Corp	MIN	WDS	TYP	CTX	TIME
BNC	1	152k	5.7m	608k	13m 9s
BNC	20	68k	5.6m	588k	9m 30s
OEC	2	269k	27.5m	994k	1hr 40m
OEC	20	128k	27.3m	981k	1hr 27m
OEC	200	48k	26.7m	965k	1hr 10m
Itwac	20	137k	24.8m	1.1m	1hr 16m

Table 1: Thesaurus creation jobs and timings

- the number of triples (types) that these words occur in (TYP)
- the number of contexts (types) that these words occur in (CTX)

We have made a number of runs with different values of MIN for BNC, OEC and Itwac and present details for some representative ones in Table 1.

For the BNC, the number of partitions that the TPMMS process was divided into was usually between ten and twenty; for the OEC and ITwac it was around 200.

For the OEC, the heuristic came into play and, in a typical run, 25 high-frequency, low-salience contexts did not play a role in the theasurus computation. They included: *modifier—more; modifier—not; object-of—have; subject-of—have.* In Gorman and Curran, increases in speed were made at substantial cost to accuracy. Here, data from these high-frequency contexts makes negligible impact on thesaurus entries.

1.3 Available thesauruses

Thesauruses of the kind described are publicly available on the Sketch Engine server (http://www.sketchengine.co.uk) based on corpora of between 50M and 2B words for, at time of writing, Chinese, English, French, Italian, Japanese, Portuguese, Slovene and Spanish.

2 Histograms for presenting statistical facts about a word's grammar

75% of the occurrences of the English noun *constraint* in the BNC are in the plural. Many dictionaries note that some nouns are usually plural: the question here is, how salient is the fact about *constraint*?[4][5]

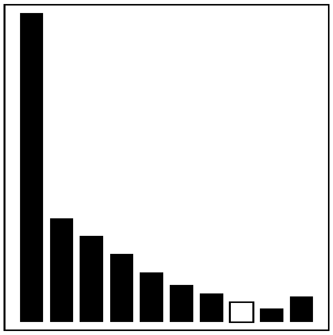


Figure 1: Distribution of nouns with respect to proportion of instances in plural, from 0 to 1 in 10 steps, with the class that *constraint* is in, in white.

To address it we need to know not only the proportion for *constraint* but also the proportion for nouns in general. If the average, across nouns, is 50% then it is probably not noteworthy. But if the average is 2%, it is. If it is 30%, we may want to ask a more specific question: for what proportion of nouns is the percentage higher than 75%. We need to view "75% plural" in the context of the whole distribution.

All the information is available. We can determine, in a large corpus such as the BNC, for each noun lemma with more than (say) fifty occurrences, what percentage is plural. We present the data in a histogram: we count the nouns for which the proportion is between 0 and 0.1, 0.1 and 0.2, . . . , 0.9 and 1. The histogram is shown in Fig 1, based on the 14,576 nouns with fifty or more occurrences in the BNC. (The first column corresponds to 6113 items.) We mark the category containing the item of interest, in red (white in this paper). We believe this is an intuitive and easy-to-interpret way of presenting a word's relative frequency in a particular grammatical context, against the background of how other words of the same word class behave.

We have implemented histograms like these in the Sketch Engine for a range of word classes and grammatical contexts. The histograms are integrated into

[4] Other 75% plural nouns which might have served as the example include: *activist bean convulsion ember feminist intricacy joist mechanic relative sandbag shutter siding teabag testicle trinket tusk.* The list immediately suggests a typology of usually-plural nouns, indicating how this kind of analysis provokes new questions.

[5] Of course plurals may be salient for one sense but not others.

the word sketch[6] for each word. (Up until now the information has been available but hard to interpret.) In accordance with the word sketch principle of not wasting screen space, or user time, on uninteresting facts, histograms are only presented where a word is in the top (or bottom) percentile for a grammatical pattern or construction.

Similar diagrams have been used for similar purposes by (Lieber and Baayen, 1997). This is, we believe, the first time that they have been offered as part of a corpus query tool.

3 Text type, subcorpora and keywords

Where a corpus has components of different text types, users often ask: "what words are distinctive of a particular text type", "what are the keywords?".[7] Computations of this kind often give unhelpful results because of the 'lumpiness' of word distributions: a word will often appear many times in an individual text, so statistics designed to find words which are distinctively different between text types will give high values for words which happen to be the topic of just one particular text (Church, 2000). (Hlaváčová and Rychlý, 1999) address the problem through defining "average reduced frequency" (ARF), a modified frequency count in which the count is reduced according to the extent to which occurrences of a word are bunched together.

The Sketch Engine now allows the user to prepare keyword lists for any subcorpus, either in relation to the full corpus or in relation to another subcorpus, using a statistic of the user's choosing and basing the result either on raw frequency or on ARF.

Acknowledgements

This work has been partly supported by the Academy of Sciences of Czech Republic under the project T100300419, by the Ministry of Education of Czech Republic within the Center of basic research LC536 and in the National Research Programme II project 2C06009.

References

Marco Baroni and Adam Kilgarriff. 2006. Large linguistically-processed web corpora for multiple languages. In *EACL*.

Tony Berber Sardinha. 2000. Comparing corpora with wordsmith tools: how large must the reference corpus be? In *Proceedings of the ACL Workshop on Comparing Corpora*, pages 7–13.

Kenneth Ward Church. 2000. Empirical estimates of adaptation: The chance of two noriegas is closer to p/2 than p2. In *COLING*, pages 180–186.

James Curran. 2004. *From Distributional to Semantic Similarity*. Ph.D. thesis, Edinburgh Univesity.

Walter Daelemans, Antal van den Bosch, and Jakub Zavrel. 1999. Forgetting exceptions is harmful in language learning. *Machine Learning*, 34(1-3).

James Gorman and James R. Curran. 2006. Scaling distributional similarity to large corpora. In *ACL*.

Gregory Grefenstette. 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer.

Jaroslava Hlaváčová and Pavel Rychlý. 1999. Dispersion of words in a language corpus. In *Proc. TSD (Text Speech Dialogue)*, pages 321–324.

Adam Kilgarriff, Pavel Rychlý, Pavel Smrž, and David Tugwell. 2004. The sketch engine. In *Proc. EURALEX*, pages 105–116.

Rochelle Lieber and Harald Baayen. 1997. Word frequency distributions and lexical semantics. *Computers in the Humanities*, 30:281–291.

Dekang Lin. 1998. Automatic retrieval and clustering of similar words. In *COLING-ACL*, pages 768–774.

Deepak Ravichandran, Patrick Pantel, and Eduard H. Hovy. 2005. Randomized algorithms and nlp: Using locality sensitive hash functions for high speed noun clustering. In *ACL*.

Karen Spärck Jones. 1964. *Synonymy and Semantic Classificiation*. Ph.D. thesis, Edinburgh University.

Julie Weeds and David J. Weir. 2005. Co-occurrence retrieval: A flexible framework for lexical distributional similarity. *Computational Linguistics*, 31(4):439–475.

[6] A word sketch is a one-page corpus-derived account of a word's grammatical and collocation behaviour.

[7] The well-established WordSmith corpus tool (http://www.lexically.net/wordsmith) has a keywords function which has been very widely used, see e.g., (Berber Sardinha, 2000).

Semantic enrichment of journal articles using chemical named entity recognition

Colin R. Batchelor
Royal Society of Chemistry
Thomas Graham House
Milton Road
Cambridge
UK CB4 0WF
batchelorc@rsc.org

Peter T. Corbett
Unilever Centre for Molecular Science Informatics
University Chemical Laboratory
Lensfield Road
Cambridge
UK CB2 1EW
ptc24@cam.ac.uk

Abstract

We describe the semantic enrichment of journal articles with chemical structures and biomedical ontology terms using Oscar, a program for chemical named entity recognition (NER). We describe how Oscar works and how it can been adapted for general NER. We discuss its implementation in a real publishing workflow and possible applications for enriched articles.

1 Introduction

The volume of chemical literature published has exploded over the past few years. The crossover between chemistry and molecular biology, disciplines which often study similar systems with contrasting techniques and describe their results in different languages, has also increased. Readers need to be able to navigate the literature more effectively, and also to understand unfamiliar terminology and its context. One relatively unexplored method for this is semantic enrichment. Substructure and similarity searching for chemical compounds is a particularly exciting prospect.

Enrichment of the bibliographic data in an article with hyperlinked citations is now commonplace. However, the actual scientific content has remained largely unenhanced, this falling to secondary services and experimental websites such as GoPubMed (Delfs *et al.*, 2005) or EBIMed (Rebholz-Schuhmann *et al.*, 2007). There are a few examples of semantic enrichment on small (a few dozen articles per year) journals such as *Nature Chemical Biology* being an example, but for a larger journal it is impractical to do this entirely by hand.

This paper concentrates on implementing semantic enrichment of journal articles as part of a publishing workflow, specifically chemical structures and biomedical terms. In the Motivation section, we introduce Oscar as a system for chemical NER and recognition of ontology terms. In the Implementation section we will discuss how Oscar works and how to set up ontologies for use with Oscar, specifically GO. In the Case study section we describe how the output of Oscar can be fed into a publishing workflow. Finally we discuss some outstanding ambiguity problems in chemical NER. We also compare the system to EBIMed (Rebholz-Schuhmann *et al.*, 2007) throughout.

2 Motivation

There are three routes for getting hold of chemical structures from chemical text—from chemical compound names, from author-supplied files containing connection tables, and from images. The preferred representation of chemical structures is as diagrams, often annotated with curly arrows to illustrate the mechanisms of chemical reactions. The structures in these diagrams are typically given numbers, which then appear in the text in bold face. However, because text-processing is more advanced in this regard than image-processing, we shall concentrate on NER, which is performed with a system called Oscar. A preliminary overview of the system was presented by Corbett and Murray-Rust (2006). Oscar is open source and can be downloaded from `http://oscar3-chem.sourceforge.net/`

As a first step in representing biomedical content, we identify Gene Ontology (GO) terms in full text.[1] (The Gene Ontology Consortium, 2000) We have chosen a relatively simple starting point in order to gain experience in implementing useful semantic markup in a publishing workflow without a substantial word-sense disambiguation effort. GO terms are largely compositional (Mungall, 2004), hence incomplete matches will still be useful, and that there is generally a low level of semantic ambiguity. For example, there are only 133 single-word GO terms, which significantly reduces the chance of polysemy for the 20000 or so others. In contrast, gene and protein

[1] We also use other OBO ontologies, specifically those for nucleic acid sequences (SO) and cell type (CL).

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 45–48,
Prague, June 2007.

(.*) activity$	→	(\1)
(.*) formation$	→	∅
(.*) synthesis$	→	∅
ribonuclease	→	RNAse
	→	ribonuclease
ˆalpha- (*etc.*)	→	α- (*etc.*)
	→	alpha- (*etc.*)
pluralize nouns		
stopwords	→	∅

Table 1: Example rules from 'Lucinda', used for generating recogniser input from OBO files

names are generally short, non-compositional and often polysemous with ordinary English words such as Cat or Rat.

3 Implementation

Oscar is intended to be a component in larger workflows, such as the Sciborg system (Copestake *et al.*, 2006). It is a shallow named-entity recogniser and does not perform deeper parsing. Hence there is no analysis of the text above the level of the term, with the exception of acronym matching, which is dealt with below, and some treatment of the boldface chemical compound numbers where they appear in section headings. It is optimized for chemical NER, but can be extended to handle general term recognition. The EBIMed system, in contrast, is a pipeline, and lemmatizes words as part of a larger workflow.

To identify plurals and other variants of non-chemical NEs we have a ruleset, nicknamed Lucinda, outlined in Table 1, for generating the input for the recogniser from external data. We use the plain-text OBO 1.2 format, which is the definitive format for the dissemination of the OBO ontologies.

We strive to keep this ruleset as small as possible, with the exception of determining plurals and a few other regular variants. The reason for keeping plurals outside the ontology is that plurals in ordinary text and in ontologies can have quite different meanings.

There is also a short stopword list applied at this stage, which is different from Oscar's internal stopword handling, described below.

3.1 Named entity recognition and resolution

Oscar has a recogniser to identify chemical names and ontology terms, and a resolver which matches NEs to ontology IDs or chemical structures. The recogniser classifies NEs according to the scheme in Corbett *et al.* (2007). The classes which are relevant here are CM, which identifies a chemical compound, either because it appears in Oscar's chemical dictionary, which also contains structures and InChIs,[2] or according to Oscar's n-gram model, regular expressions and other heuristics and ASE, a single word ending in "-ase" or "-ases" and representing an enzyme type. We add the class ONT to these, to cover terms found in ontologies that do not belong in the other classes, and STOP, which is the class of stopwords.

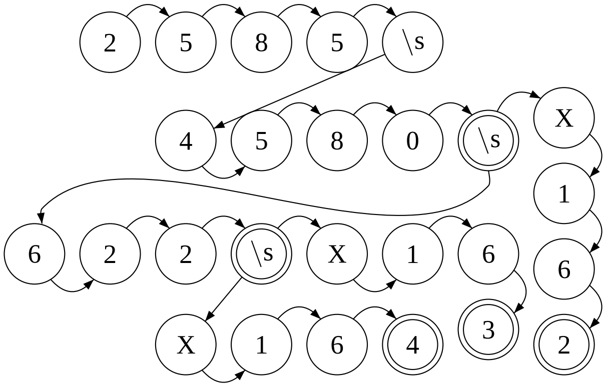


Figure 1: Cartoon of part of the recogniser. The mapping between this automaton and example GO terms is given in Table 2.

GO term	Regex pair
bud neck	2585\s4580\s
	2585\s4580\sX162
bud neck polarisome	2585\s4580\s622\s
	2585\s4580\s622\sX163
polarisome	622\s
	622\sX164

Table 2: Mapping in Fig. 1. The regexes are purely illustrative. IDs 162, 163 and 164 map on to GO:0005935, GO:0031560 and GO:0000133 respectively.

We sketch the recogniser in Fig. 1. To build the recogniser: Each term in the input data is tokenized and the tokens converted into a sequence of digits followed by a space. These new tokens are concatenated and converted into a pair of regular expressions. One of these expressions has X followed by a term ID appended to it. These regex–regex pairs are converted into finite automata, the union of which is determinized. The resulting DFA is examined for accept states. For each accept state for which a transition to X is also present, the sequences of digits after the X is used to build a mapping of accept states to ontology IDs (Table 2).

To apply the recogniser: The input text is tokenized, and for each token a set of representations is calculated which map to sequences of digits as above. We then make an empty set of DFA instances (a pointer to the DFA,

[2] An InChI is a canonical identifier for a chemical compound. http://www.iupac.org/inchi/

which state it's in and which tokens it has matched so far), and for each token, add a new DFA instance for each DFA, and for each representation of the token, clone the DFA instance. If it does not accept the digit-sequence representation of the token, throw it away. If it is in an accept state, note which tokens it has matched, and if the accept state maps to an ontology ID (ontID), we have an NE which can be annotated with the ontID.

Take all of the potential NEs. For all NEs that have the same sequence of tokens, share all of the ontIDs. Assign its class according to a priority list where `STOP` comes first and `CM` precedes `ASE` and `ONT`. For the system in Fig. 1, the phrase "bud neck polarisome" matches three IDs. We choose the longest–leftmost sequence. If the resolver generates an InChI for an NE, we look up this InChI in ChEBI (de Matos *et al.*, 2006), a biochemical ontology, and take the ontology ID. This has the effect of aligning ChEBI with other databases and systematic nomenclature.

3.2 Gene Ontology

In working out how to mine the literature for GO terms, we have taken our lead from the domain experts, the GO editors and the curators of the Gene Ontology Annotation (GOA) database.

The Functional Curation task in the first BioCreative exercise (Blaschke *et al.*, 2005) is the closest we have found to a systematic evaluation of GO term identification. The brief was to assign GO annotations to human proteins and recover supporting text. The GOA curators evaluated the results (Camon *et al.*, 2005) and list some common mistakes in the methods used to identify GO terms. These include annotating to obsolete terms, predicting GO terms on too tenuous a link with the original text, for example in one case the phrase "pH value" was annotated to "pH domain binding" (GO:0042731), difficulties with word order, and choosing too much supporting text, for example an entire first paragraph of text.

So at the suggestion of the GO editors, Oscar works on exact matches to term names (as preprocessed above) and their exact (within the OBO syntax) synonyms.

The most relevant GO terms to chemistry concern enzymes, which are proteins that catalyse chemical processes. Typically their names are multiword expressions ending in "-ase". The enzyme A B Xase will often be represented by GO terms "A B Xase activity", a description of what the enzyme does, and "A B Xase complex", a cellular component which consists of two or more protein subunits. In general the bare phrase "A B Xase" will refer to the activity, so the ruleset in Table 1 deletes the word "activity" from the GO term.

We shall briefly compare our method with the algorithms in EBIMed and GoPubMed. The EBIMed algorithm for GO term identification is very similar to ours, except for the point about lemmatization listed above, and its explicit variation of character case, which is handled in Oscar by its case normalization algorithm. In contrast, the algorithm in GoPubMed works by matching short 'seed' terms and then expanding them. This copes with cases such as "protein threonine/tyrosine kinase activity" (GO:0030296) where the full term is unlikely to be found in ordinary text; the words "protein" and "activity" are generally omitted. However, the approach in (Delfs *et al.*, 2005) cannot be applied blindly; the authors claim for example that "biosynthesis" can be ignored without compromising the reader's understanding. In chemistry journal articles most mentions of a chemical compound will not refer to how it is formed in nature; they will refer to the compound itself, its analogues or other processes. In fact, our ruleset in Table 1 explicitly disallows GO term synonyms ending in " synthesis" or " formation" since they do not necessarily represent biological processes. It is also not clear from Delfs *et al.* (2005) how robust the algorithm is to the sort of errors identified by Camon *et al.* (2005).

4 Case study

The problem is to take a journal article, apply meaningful and useful annotations, connect them to stable resources, allow technical editors to check and add further annotations, and disseminate the article in enriched form.

Most chemical publishers use XML as a stable format for maintaining their documents for at least some stages of the publication process. The Sciborg project (Copestake *et al.*, 2006) and the Royal Society of Chemistry (RSC) use SciXML (Rupp *et al.*, 2006) and RSC XML respectively. For the overall Sciborg workflow, standoff annotation is used to store the different sets of annotations. For the purposes of this paper, however, we make use of the inline output of Oscar, which is SciXML with `<ne>` elements for the annotations.

Not all of the RSC XML need be mined for NEs; much of it is bibliographic markup which can confuse parsers. Only the useful parts are converted into SciXML and passed to Oscar, where they are annotated. These SciXML annotations are then pasted back into the RSC XML, where they can be checked by technical editors. In running text, NEs are annotated with an ID local to the XML file, which refers to `<compound>` and `<annotation>` elements in a block at the end, which contain chemical structure information and ontology IDs. This is a lightweight compromise between pure standoff and pure inline annotation.

We find useful annotations by aggressive thresholding. The only classes which survive are `ONT`s, and those `CM`s which have a chemical structure found by the resolver. This enables the chemical NER part of Oscar to be tuned for high recall even as part of a publishing workflow. Only CMs which correspond to an unambiguous molecule or molecular ion are treated as a chemical compound; everything else is referred to an appropriate ontology. We use the InChI as a stable representation for chemical structure, and the curated OBO ontologies for biomedical terms.

The role of technical editors is to remove faulty annotations, add new compounds to the chemical dictionary, based on chemical structures supplied by authors, suggest new GO terms to the ontology curators, and extend the stopword lists of both Oscar and Lucinda as appropriate. At present (May 2007), this happens after publication of articles on the web, but is intended to become part of the routine editing process in the course of 2007.

This enriched XML can then be converted into HTML and RSS by means of XSL stylesheets and database lookups, as in the RSC's Project Prospect.[3] The immediate benefits of this work are increased readability of articles for readers and extensive cross-linking with other articles that have been enhanced in the same way. Future developments could easily involve structure-based searching, ontology-based search of journal articles, and finding correlations between biological processes and small molecule structures.

5 Ambiguity in chemical NER

One important omission is disambiguating the exact referent of a chemical name, which is not always clear without context. For example "the pyridine **6**", is a class description, but the phrase "the pyridine molecule" refers to a particular compound. ChEBI, which contains an ontology of molecular structure, uses plurals to indicate chemical classes, for example "benzenes", which is often, but not always, what "benzenes" means in text. Currently Oscar does not distinguish between singular and plural.

Amino acids and saccharides are particularly troublesome on account of homochirality. Unless otherwise specified, "histidine" and "ribose" specify the molecules with the chirality found in nature, or to be precise, L-histidine and D-ribose respectively. What is even worse is that "histidine" seldom refers to the independent molecule; it usually means the histidine residue, part of a larger entity.

6 Acknowledgements

We thank Dietrich Rebholz-Schuhmann for useful discussions. CRB thanks Jane Lomax, Jen Clark, Amelia Ireland and Midori Harris for extensive cooperation and help, and Richard Kidd, Neil Hunter and Jeff White at the RSC. PTC thanks Ann Copestake and Peter Murray-Rust for supervision. This work was funded by EPSRC (EP/C010035/1).

References

Christian Blaschke, Eduardo Andres Leon, Martin Krallinger and Alfonso Valencia. 2005. Evaluation of BioCreAtIvE assessment of task 2 *BMC Bioinformatics* 6(Suppl 1):S16

Evelyn B. Camon, Daniel G. Barrell, Emily C. Dimmer, Vivian Lee, Michele Magrane, John Maslen, David Binns and Rolf Apweiler. 2005. An evaluation of GO annotation retrieval for BioCreAtIvE and GOA *BMC Bioinformatics* 6(Suppl 1):S17

Ann Copestake, Peter Corbett, Peter Murray-Rust, C. J. Rupp, Advaith Siddharthan, Simone Teufel and Ben Waldron. 2006. An Architecture for Language Technology for Processing Scientific Texts. In Proceedings of the 4th UK E-Science All Hands Meeting. Nottingham, UK.

Peter Corbett, Colin Batchelor and Simone Teufel. 2007. Annotation of Chemical Named Entities. In Proceedings of BioNLP in ACL (BioNLP'07).

Peter T. Corbett and Peter Murray-Rust. 2006. High-throughput identification of chemistry in life science texts. *LNCS*, 4216:107–118.

P. de Matos, M. Ennis, M. Darsow, M. Guedj, K. Degtyarenko, and R. Apweiler. 2006. ChEBI - Chemical Entities of Biological Interest *Nucleic Acids Research*, Database Summary Paper 646.

The Gene Ontology Consortium. 2000. Gene Ontology: Tool for the Unification of Biology *Nature Genetics*, 25:25–29.

Ralph Delfs, Andreas Doms, Alexander Kozlenkov and Michael Schroeder. 2004. GoPubMed: Exploring PubMed with the GeneOntology. Proceedings of German Bioinformatics Conference, 169–178.

Christopher J. Mungall. 2004. Obol: integrating language and meaning in bio-ontologies. *Comparative and Functional Genomics*, 5:509–520.

Dietrich Rebholz-Schuhmann, Harald Kirsch, Miguel Arregui, Sylvain Gaudan, Mark Riethoven and Peter Stoehr. 2007. EBIMed—text crunching to gather facts for proteins from Medline. *Bioinformatics*, 23(2):e237–e244.

C. J. Rupp, Ann Copestake, Simone Teufel and Benjamin Waldron. 2006. Flexible Interfaces in the Application of Language Technology to an eScience Corpus. In Proceedings of the 4th UK E-Science All Hands Meeting. Nottingham, UK.

[3] http://www.projectprospect.org/

An API for Measuring the Relatedness of Words in Wikipedia

Simone Paolo Ponzetto and **Michael Strube**
EML Research gGmbH
Schloss-Wolfsbrunnenweg 33
69118 Heidelberg, Germany
http://www.eml-research.de/nlp

Abstract

We present an API for computing the semantic relatedness of words in Wikipedia.

1 Introduction

The last years have seen a large amount of work in Natural Language Processing (NLP) using measures of semantic similarity and relatedness. We believe that the extensive usage of such measures derives also from the availability of robust and freely available software that allows to compute them (Pedersen et al., 2004, WordNet::Similarity).

In Ponzetto & Strube (2006) and Strube & Ponzetto (2006) we proposed to take the Wikipedia categorization system as a semantic network which served as basis for computing the semantic relatedness of words. In the following we present the API we used in our previous work, hoping that it will encourage further research in NLP using Wikipedia[1].

2 Measures of Semantic Relatedness

Approaches to measuring semantic relatedness that use lexical resources transform these resources into a network or graph and compute relatedness using paths in it (see Budanitsky & Hirst (2006) for an extensive review). For instance, Rada et al. (1989) traverse MeSH, a term hierarchy for indexing articles in Medline, and compute semantic relatedness straightforwardly in terms of the number of edges between terms in the hierarchy. Jarmasz & Szpakowicz (2003) use the same approach with *Roget's Thesaurus* while Hirst & St-Onge (1998) apply a similar strategy to WordNet.

3 The Application Programming Interface

The API computes semantic relatedness by:

1. taking a pair of **words as input**;
2. **retrieving the Wikipedia articles** they refer to (via a disambiguation strategy based on the link structure of the articles);
3. **computing paths in the Wikipedia categorization graph** between the categories the articles are assigned to;
4. **returning as output the set of paths found, scored** according to some measure definition.

The implementation includes *path-length* (Rada et al., 1989; Wu & Palmer, 1994; Leacock & Chodorow, 1998), *information-content* (Resnik, 1995; Seco et al., 2004) and *text-overlap* (Lesk, 1986; Banerjee & Pedersen, 2003) measures, as described in Strube & Ponzetto (2006).

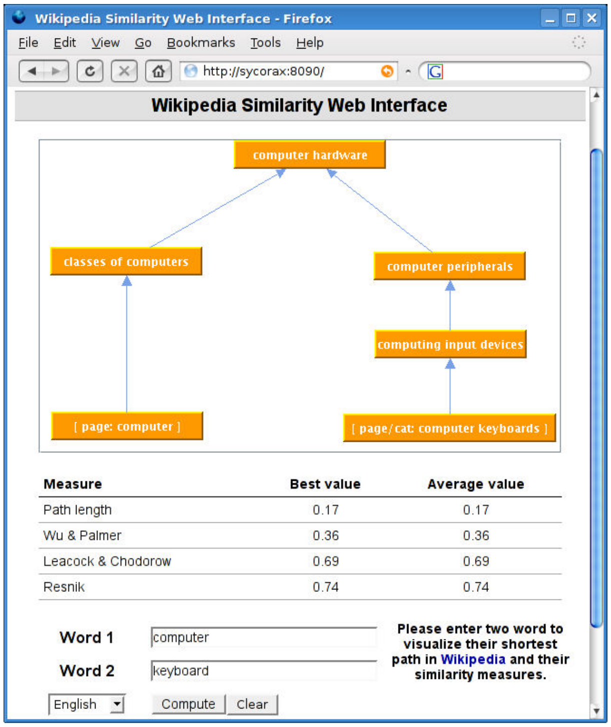
The API is built on top of several modules and can be used for tasks other than Wikipedia-based relatedness computation. On a basic usage level, it can be used to retrieve Wikipedia articles by name, optionally using disambiguation patterns, as well as to find a ranked set of articles satisfying a search query (via integration with the Lucene[2] text search engine). Additionally, it provides functionality for visualizing the computed paths along the Wikipedia categorization graph as either Java Swing components or applets (see Figure 1), based on the JGraph library[3], and methods for computing centrality scores of the Wikipedia categories using the PageRank algorithm (Brin & Page, 1998). Finally, it currently

[1] The software can be freely downloaded at http://www.eml-research.de/nlp/download/wikipediasimilarity.php.

[2] http://lucene.apache.org

[3] http://www.jgraph.com

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 49–52,
Prague, June 2007. ©2007 Association for Computational Linguistics


Wikipedia Similarity Web Interface - Firefox
File Edit View Go Bookmarks Tools Help
http://sycorax:8090/
Wikipedia Similarity Web Interface
computer hardware
classes of computers
computer peripherals
computing input devices
[page: computer]
[page/cat: computer keyboards]
English
Compute
Clear

Measure	Best value	Average value
Path length	0.17	0.17
Wu & Palmer	0.36	0.36
Leacock & Chodorow	0.69	0.69
Resnik	0.74	0.74

Word 1: computer

Word 2: keyboard

Please enter two word to visualize their shortest path in Wikipedia and their similarity measures.

Figure 1: Shortest path between computer and keyboard in the English Wikipedia.

provides multilingual support for the English, German, French and Italian Wikipedias and can be easily extended to other languages[4].

4 Software Architecture

Wikipedia is freely available for download, and can be accessed using robust Open Source applications, e.g. the MediaWiki software[5], integrated within a Linux, Apache, MySQL and PHP (LAMP) software bundle. The architecture of the API consists of the following modules:

1. **RDBMS**: at the lowest level, the encyclopedia content is stored in a relational database management system (e.g. MySQL).
2. **MediaWiki**: a suite of PHP routines for interacting with the RDBMS.
3. **WWW-Wikipedia Perl library**[6]: responsible for querying MediaWiki, parsing and structuring the returned encyclopedia pages.
4. **XML-RPC server**: an intermediate communication layer between Java and the Perl routines.
5. **Java wrapper library**: provides a simple interface to create and access the encyclopedia page objects and compute the relatedness scores.

The information flow of the API is summarized by the sequence diagram in Figure 2. The higher input/output layer the user interacts with is provided by a Java API from which Wikipedia can be queried. The Java library is responsible for issuing HTTP requests to an XML-RPC daemon which provides a layer for calling Perl routines from the Java API. Perl routines take care of the bulk of querying encyclopedia entries to the MediaWiki software (which in turn queries the database) and efficiently parsing the text responses into structured objects.

5 Using the API

The API provides factory classes for querying Wikipedia, in order to retrieve encyclopedia entries as well as relatedness scores for word pairs. In practice, the Java library provides a simple programmatic interface. Users can accordingly access the library using only a few methods given in the factory classes, e.g. `getPage(word)` for retrieving Wikipedia articles titled `word` or `getRelatedness(word1,word2)`, for computing the relatedness between `word1` and `word2`, and `display(path)` for displaying a path found between two Wikipedia articles in the categorization graph. Examples of programmatic usage of the API are presented in Figure 3. In addition, the software distribution includes UNIX shell scripts to access the API interactively from a terminal, i.e. it does not require any knowledge of Java.

6 Application scenarios

Semantic relatedness measures have proven useful in many NLP applications such as word sense disambiguation (Kohomban & Lee, 2005; Patwardhan et al., 2005), information retrieval (Finkelstein et al., 2002), information extraction pattern induction (Stevenson & Greenwood, 2005), interpretation of noun compounds (Kim & Baldwin, 2005), para-

[4] In contrast to WordNet::Similarity, which due to the structural variations between the respective wordnets was reimplemented for German by Gurevych & Niederlich (2005).

[5] `http://www.mediawiki.org`

[6] `http://search.cpan.org/dist/WWW-Wikipedia`

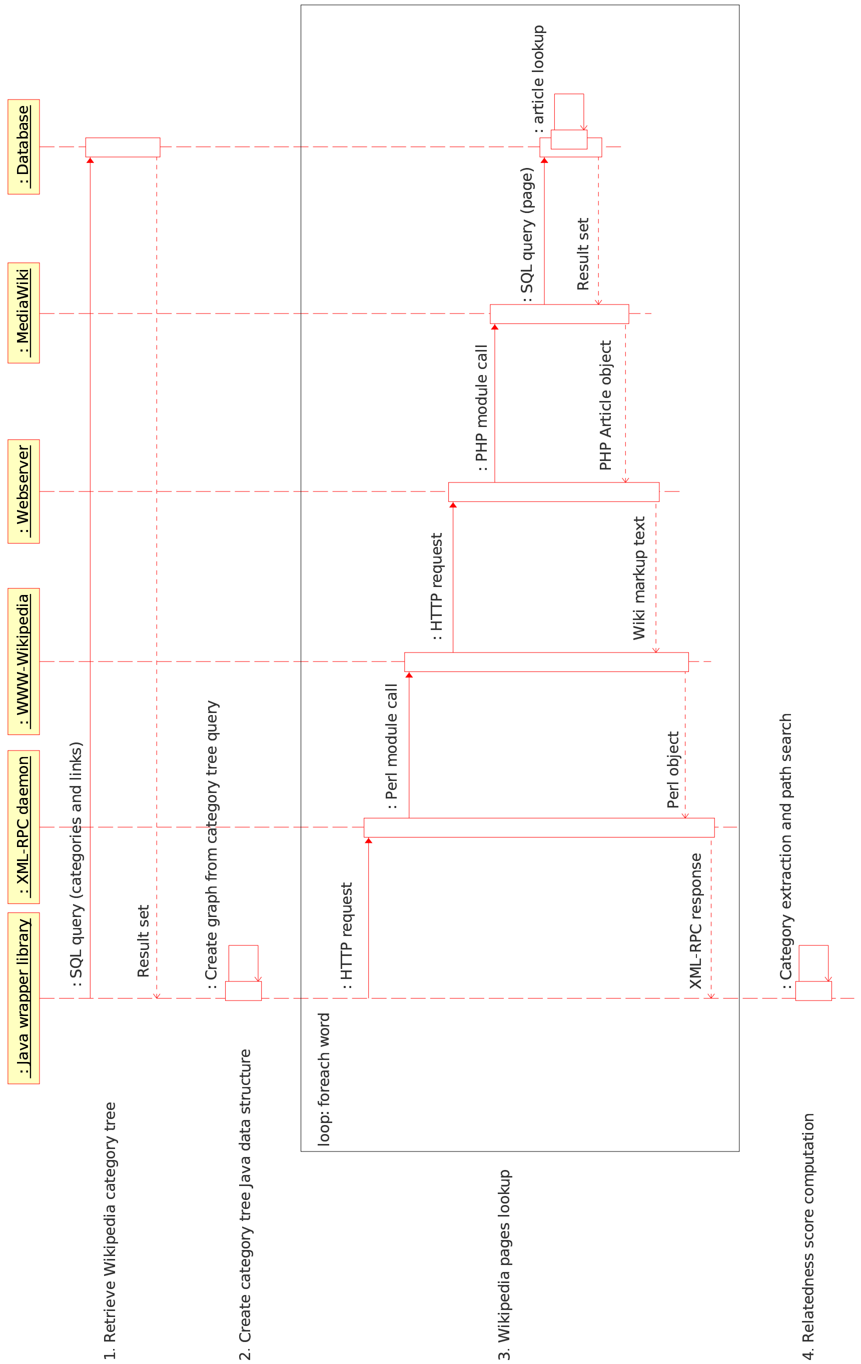


Figure 2: API processing sequence diagram. Wikipedia pages and relatedness measures are accessed through a Java API. The wrapper communicates with a Perl library designed for Wikipedia access and parsing through an XML-RPC server. WWW-Wikipedia in turn accesses the database where the encyclopedia is stored by means of appropriate queries to MediaWiki.

```
// 1. Get the English Wikipedia page titled "King" using "chess" as disambiguation
WikipediaPage page = WikipediaPageFactory.getInstance().getWikipediaPage("King","chess");

// 2. Get the German Wikipedia page titled "Ufer" using "Kueste" as disambiguation
WikipediaPage page = WikipediaPageFactory.getInstance().getWikipediaPage("Ufer","Kueste",Language.DE);

// 3a. Get the Wikipedia-based path-length relatedness measure between "computer" and "keyboard"
WikiRelatedness relatedness = WikiRelatednessFactory.getInstance().getWikiRelatedness("computer","keyboard");
double shortestPathMeasure = relatedness.getShortestPathMeasure();
// 3b. Display the shortest path
WikiPathDisplayer.getInstance().display(relatedness.getShortestPath());

// 4. Score the importance of the categories in the English Wikipedia using PageRank
WikiCategoryGraph<DefaultScorableGraph<DefaultEdge>> categoryTree =
        WikiCategoryGraphFactory.getCategoryGraphForLanguage(Language.EN);
categoryTree.getCategoryGraph().score(new PageRank());
```

Figure 3: Java API sample usage.

phrase detection (Mihalcea et al., 2006) and spelling correction (Budanitsky & Hirst, 2006). Our API provides a flexible tool to include such measures into existing NLP systems while using Wikipedia as a knowledge source. Programmatic access to the encyclopedia makes also available in a straightforward manner the large amount of structured text in Wikipedia (e.g. for building a language model), as well as its rich internal link structure (e.g. the links between articles provide phrase clusters to be used for query expansion scenarios).

Acknowledgements: This work has been funded by the Klaus Tschira Foundation, Heidelberg, Germany. The first author has been supported by a KTF grant (09.003.2004). We thank our colleagues Katja Filippova and Christoph Müller for helpful feedback.

References

Banerjee, S. & T. Pedersen (2003). Extended gloss overlap as a measure of semantic relatedness. In *Proc. of IJCAI-03*, pp. 805–810.

Brin, S. & L. Page (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1–7):107–117.

Budanitsky, A. & G. Hirst (2006). Evaluating WordNet-based measures of semantic distance. *Computational Linguistics*, 32(1).

Finkelstein, L., E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan, G. Wolfman & E. Ruppin (2002). Placing search in context: The concept revisited. *ACM Transactions on Information Systems*, 20(1):116–131.

Gurevych, I. & H. Niederlich (2005). Accessing GermaNet data and computing semantic relatedness. In *Comp. Vol. to Proc. of ACL-05*, pp. 5–8.

Hirst, G. & D. St-Onge (1998). Lexical chains as representations of context for the detection and correction of malapropisms. In C. Fellbaum (Ed.), *WordNet: An Electronic Lexical Database*, pp. 305–332. Cambridge, Mass.: MIT Press.

Jarmasz, M. & S. Szpakowicz (2003). Roget's Thesaurus and semantic similarity. In *Proc. of RANLP-03*, pp. 212–219.

Kim, S. N. & T. Baldwin (2005). Automatic interpretation of noun compounds using WordNet similarity. In *Proc. of IJCNLP-05*, pp. 945–956.

Kohomban, U. S. & W. S. Lee (2005). Learning semantic classes for word sense disambiguation. In *Proc. of ACL-05*, pp. 34–41.

Leacock, C. & M. Chodorow (1998). Combining local context and WordNet similarity for word sense identification. In C. Fellbaum (Ed.), *WordNet. An Electronic Lexical Database*, Chp. 11, pp. 265–283. Cambridge, Mass.: MIT Press.

Lesk, M. (1986). Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual Conference on Systems Documentation,* Toronto, Ontario, Canada, pp. 24–26.

Mihalcea, R., C. Corley & C. Strapparava (2006). Corpus-based and knowledge-based measures of text semantic similarity. In *Proc. of AAAI-06*, pp. 775–780.

Patwardhan, S., S. Banerjee & T. Pedersen (2005). SenseRelate::TargetWord – A generalized framework for word sense disambiguation. In *Proc. of AAAI-05*.

Pedersen, T., S. Patwardhan & J. Michelizzi (2004). WordNet::Similarity – Measuring the relatedness of concepts. In *Comp. Vol. to Proc. of HLT-NAACL-04*, pp. 267–270.

Ponzetto, S. P. & M. Strube (2006). Exploiting semantic role labeling, WordNet and Wikipedia for coreference resolution. In *Proc. of HLT-NAACL-06*, pp. 192–199.

Rada, R., H. Mili, E. Bicknell & M. Blettner (1989). Development and application of a metric to semantic nets. *IEEE Transactions on Systems, Man and Cybernetics*, 19(1):17–30.

Resnik, P. (1995). Using information content to evaluate semantic similarity in a taxonomy. In *Proc. of IJCAI-95*, Vol. 1, pp. 448–453.

Seco, N., T. Veale & J. Hayes (2004). An intrinsic information content metric for semantic similarity in WordNet. In *Proc. of ECAI-04*, pp. 1089–1090.

Stevenson, M. & M. Greenwood (2005). A semantic approach to IE pattern induction. In *Proc. of ACL-05*, pp. 379–386.

Strube, M. & S. P. Ponzetto (2006). WikiRelate! Computing semantic relatedness using Wikipedia. In *Proc. of AAAI-06*, pp. 1419–1424.

Wu, Z. & M. Palmer (1994). Verb semantics and lexical selection. In *Proc. of ACL-94*, pp. 133–138.

Deriving an Ambiguous Word's Part-of-Speech Distribution from Unannotated Text

Reinhard Rapp
Universitat Rovira i Virgili
Pl. Imperial Tarraco, 1
E-43005 Tarragona, Spain
reinhard.rapp@urv.cat

Abstract

A distributional method for part-of-speech induction is presented which, in contrast to most previous work, determines the part-of-speech distribution of syntactically ambiguous words without explicitly tagging the underlying text corpus. This is achieved by assuming that the word pair consisting of the left and right neighbor of a particular token is characteristic of the part of speech at this position, and by clustering the neighbor pairs on the basis of their middle words as observed in a large corpus. The results obtained in this way are evaluated by comparing them to the part-of-speech distributions as found in the manually tagged Brown corpus.

1 Introduction

The purpose of this study is to automatically induce a system of word classes that is in agreement with human intuition, and then to assign all possible parts of speech to a given ambiguous or unambiguous word. Two of the pioneering studies concerning this as yet not satisfactorily solved problem are Finch (1993) and Schütze (1993) who classify words according to their context vectors as derived from a corpus. More recent studies try to solve the problem of POS induction by combining distributional and morphological information (Clark, 2003; Freitag, 2004), or by clustering words and projecting them to POS vectors (Rapp, 2005).

Whereas all these studies are based on global co-occurrence vectors who reflect the overall behavior of a word in a corpus, i.e. who in the case of syntactically ambiguous words are based on POS-mixtures, in this paper we raise the question if it is really necessary to use an approach based on mixtures or if there is some way to avoid the mixing beforehand. For this purpose, we suggest to look at local contexts instead of global co-occurrence vectors. As can be seen from human performance, in almost all cases the local context of a syntactically ambiguous word is sufficient to disambiguate its part of speech.

The core assumption underlying our approach, which in the context of cognition and child language has been proposed by Mintz (2003), is that words of a particular part of speech often have the same left and right neighbors, i.e. a pair of such neighbors can be considered to be characteristic of a part of speech. For example, a noun may be surrounded by the pair "*the ... is*", a verb by the pair "*he ... the*", and an adjective by the pair "*the ... thing*". For ease of reference, in the remainder of this paper we call these local contexts *neighbor pairs*. The idea is now to cluster the neighbor pairs on the basis of the middle words they occur with. This way neighbor pairs typical of the same part of speech are grouped together. For classification, a word is assigned to the cluster where its neighbor pairs are found. If its neighbor pairs are spread over several clusters, the word can be assumed to be ambiguous. This way ambiguity detection follows naturally from the methodology.

2 Approach

Let us illustrate our approach by looking at Table 1. The rows in the table are the neighbor pairs that we want to consider, and the columns are suitable middle words as we find them in a corpus. Most words in our example are syntactically unambiguous. Only *link* can be either a noun or a verb and therefore shows the co-occurrence patterns of both. Apart from the particular choice of features, what distinguishes our approach from most others is that we do not cluster the words (columns) which would be the more straightforward thing to do. Instead we cluster the neighbor pairs (rows). Clustering the columns would be fine for unambiguous words, but has the drawback that ambiguous words

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 53–56,
Prague, June 2007. ©2007 Association for Computational Linguistics

tend to be assigned only to the cluster relating to their dominant part of speech. This means that no ambiguity detection takes place at this stage.

In contrast, the problem of demixing can be avoided by clustering the rows which leads to the condensed representation as shown in Table 2. The neighbor pairs have been grouped in such a way that the resulting clusters correspond to classes that can be linguistically interpreted as nouns, adjectives, and verbs. As desired, all unambiguous words have been assigned to only a single cluster, and the ambiguous word *link* has been assigned to the two appropriate clusters.

Although it is not obvious from our example, there is a drawback of this approach. The disadvantage is that by avoiding the ambiguity problem for words we introduce it for the neighbor pairs, i.e. ambiguities concerning neighbor pairs are not resolved. Consider, for example, the neighbor pair "*then ... comes*", where the middle word can either be a personal pronoun like *he* or a proper noun like *John*. However, we believe that this is a problem that for several reasons is of less importance: Firstly, we are not explicitly interested in the ambiguities of neighbor pairs. Secondly, the ambiguities of neighbor pairs seem less frequent and less systematic than those of words (an example is the omnipresent noun/verb ambiguity in English), and therefore the risk of misclusterings is lower. Thirdly, this problem can be reduced by considering longer contexts which tend to be less ambiguous. That is, by choosing an appropriate context width a reasonable tradeoff between data sparseness and ambiguity reduction can be chosen.

	car	cup	discuss	link	quick	seek	tall	thin
a ... has	●	●		●				
a ... is	●	●		●				
a ... man					●		●	●
a ... woman					●		●	●
the ... has	●	●		●				
the ... is	●	●		●				
the ... man					●		●	●
the ... woman					●		●	●
to ... a			●	●		●		
to ... the			●	●		●		
you ... a			●	●		●		
you ... the			●	●		●		

Table 1: Matrix of neighbor pairs and their corresponding middle words.

	car	cup	discuss	link	quick	seek	tall	thin
a ... has, a ... is, the ... has, the ... is	●	●		●				
a ... man, a ... woman, the ... man, the ... woman					●		●	●
to ... a, to ... the, you ... a, you ... the			●	●		●		

Table 2: Clusters of neighbor pairs.

3 Implementation

Our computations are based on the 100 million word British National Corpus. As the number of word types and neighbor pairs is prohibitively high in a corpus of this size, we considered only a selected vocabulary, as described in section 4. From all neighbor pairs we chose the top 2000 which had the highest co-occurrence frequency with the union of all words in the vocabulary and did not contain punctuation marks.

By searching through the full corpus, we constructed a matrix as exemplified in Table 1. However, as a large corpus may contain errors and idiosyncrasies, the matrix cells were not filled with binary yes/no decisions, but with the frequency of a word type occurring as the middle word of the respective neighbor pair. Note that we used raw co-occurrence frequencies and did not apply any association measure. However, to account for the large variation in word frequency and to give an equal chance to each word in the subsequent computations, the matrix columns were normalized.

As our method for grouping the rows we used K-means clustering with the cosine coefficient as our similarity measure. The clustering algorithm was started using random initialization. In order to be able to easily compare the clustering results with expectation, the number of clusters was specified to correspond to the number of expected word classes.

After the clustering has been completed, to obtain their centroids, in analogy to Table 2 the column vectors for each cluster are summed up. The centroid values for each word can now be interpreted as evidence of this word belonging to the class described by the respective cluster. For example, if we obtained three clusters corresponding to nouns, verbs, and adjectives, and if the corresponding centroid values for e.g. the word *link* would be 0.7, 0.3, and 0.0, this could be interpreted such that in 70% of its corpus occurrences *link* has the function of a noun, in 30% of the cases it appears as a verb, and that it never occurs as an adjective. Note that the centroid values for a particular word will always add up to 1 since, as mentioned above, the column vectors have been normalized beforehand.

As elaborated in Rapp (2007), another useful application of the centroid vectors is that they allow us to judge the quality of the neighbor pairs with respect to their selectivity regarding a particular word class. If the row vector of a neighbor pair is very similar to the centroid of its cluster, then it can be assumed that this neighbor pair only accepts middle words of the correct class, whereas neighbor pairs with lower similarity to the centroid are probably less selective, i.e. they occasionally allow for words from other clusters.

4 Results

As our test vocabulary we chose a sample of 50 words taken from a previous study (Rapp, 2005). The list of words is included in Table 3 (columns 1 and 8). Columns 2 to 4 and 9 to 11 of Table 3 show the centroid values corresponding to each word after the procedure described in the previous section has been conducted, that is, the 2000 most frequent neighbor pairs of the 50 words were clustered into three groups. For clarity, all values were multiplied by 1000 and rounded.

To facilitate reference, instead of naming each cluster by a number or by specifying the corresponding list of neighbor pairs (as done in Table 2), we manually selected linguistically motivated names, namely *noun*, *verb*, and *adjective*.

If we look at Table 3, we find that some words, such as *encourage*, *imagine*, and *option*, have one value close to 1000, with the other two values in the one digit range. This is a typical pattern for unambiguous words that belong to only one word class. However, perhaps unexpectedly, the majority of words has values in the upper two digit or three digit range in two or even three columns. This means that according to our system most words seem to be ambiguous in one or another way. For example, the word *brief*, although in the majority of cases clearly an adjective in the sense of *short*, can occasionally also occur as a noun (in the sense of *document*) or a verb (in the sense of *to instruct somebody*). In other cases, the occurrences of different parts of speech are more balanced. An example is the verb *to strike* versus the noun *the strike.*

According to our judgment, the results for all words seem roughly plausible. Only the values for *rain* as a noun versus a verb seemed on first glance counterintuitive, but can be explained by the fact that for semantic reasons the verb *rain* usually only occurs in third person singular, i.e. in its inflected form *rains*.

To provide a more objective measure for the quality of the results, columns 5 to 7 and 12 to 14 of Table 3 show the occurrence frequencies of the 50 words as nouns, verbs, and adjectives in the manually POS-tagged Brown corpus, which is probably almost error free (Kuçera, & Francis, 1967). The respective tags in the Brown-tagset are NN, VB, and JJ.

Generally, the POS-distributions of the Brown corpus show a similar pattern as the automatically generated ones. For example, for *drop* the ratios of the automatically generated numbers 334 / 643 / 24 are similar to those of the pattern from the Brown corpus which is 24 / 34 / 1. Overall, for 48 of the 50 words the outcome with regard to the most likely POS is identical, with the two exceptions being the ambiguous words *finance* and *suit.* Although even in these cases the correct two parts of speech obtain the emphasis, the distribution of the weighting among them is somewhat different.

5 Summary and Future Work

A statistical approach has been presented which clusters contextual features (neighbor pairs) as observed in a large text corpus and derives syntactically oriented word classes from the clusters. In addition, for each word a probability of its occurrence as a member of each of the classes is computed.

Of course, many questions are yet to be explored, among them the following: Can a singular value decomposition (to be in effect only temporarily for the purpose of clustering) reduce the problem of data sparseness? Can biclustering (also referred to as co-clustering or two-mode clustering, i.e. the simultaneous clustering of the rows and columns of a matrix) improve results? Does the approach scale to larger vocabularies? Can it be extended to word sense induction by looking at longer distance equivalents to middle words and neighbor pairs (which could be homographs and pairs of words strongly associated to them)? All these are strands of research that we look forward to explore.

	Simulation			Brown Corpus				Simulation			Brown Corpus		
	Noun	Verb	Adj.	NN	VB	JJ		Noun	Verb	Adj.	NN	VB	JJ
accident	978	8	15	33	0	0	lunch	741	198	60	32	1	0
belief	972	17	11	64	0	0	maintain	4	993	3	0	60	0
birth	968	15	18	47	0	0	occur	15	973	13	0	43	0
breath	946	21	33	51	0	0	option	984	10	7	5	0	0
brief	132	50	819	8	0	63	pleasure	931	16	54	60	1	0
broad	59	7	934	0	0	82	protect	4	995	1	0	34	0
busy	22	22	956	0	1	56	prove	5	989	6	0	53	0
catch	71	920	9	3	39	0	quick	47	14	938	1	0	58
critical	51	13	936	0	0	57	rain	881	64	56	66	2	0
cup	957	23	21	43	1	0	reform	756	221	23	23	3	0
dangerous	37	29	934	0	0	46	rural	66	13	921	0	0	46
discuss	3	991	5	0	28	0	screen	842	126	32	42	5	0
drop	334	643	24	24	34	1	seek	8	955	37	0	69	0
drug	944	10	46	20	0	0	serve	20	958	22	0	107	0
empty	48	187	765	0	0	64	slow	43	141	816	0	8	48
encourage	7	990	3	0	46	0	spring	792	130	78	102	6	0
establish	2	995	2	0	58	0	strike	544	424	32	25	22	0
expensive	55	14	931	0	0	44	suit	200	789	11	40	8	0
familiar	42	17	941	0	0	72	surprise	818	141	41	44	5	3
finance	483	473	44	9	18	0	tape	868	109	23	31	0	0
grow	15	973	12	0	61	0	thank	14	983	3	0	35	0
imagine	4	993	4	0	61	0	thin	32	58	912	0	2	90
introduction	989	0	11	28	0	0	tiny	27	1	971	0	0	49
link	667	311	23	12	4	0	wide	9	4	988	0	0	115
lovely	41	7	952	0	0	44	wild	220	6	774	0	0	51

Table 3: List of 50 words and their values (scaled by 1000) from each of the three cluster centroids. For comparison, POS frequencies from the manually tagged Brown corpus are given.

Acknowledgments

This research was supported by a Marie Curie Intra-European Fellowship within the 6th Framework Programme of the European Community.

References

Clark, Alexander (2003). Combining distributional and morphological information for part of speech induction. *Proceedings of 10th EACL Conference*, Budapest, 59–66.

Finch, Steven (1993). *Finding Structure in Language*. PhD Thesis, University of Edinburgh.

Freitag, Dayne (2004). Toward unsupervised whole-corpus tagging. *Proc. of 20th COLING*, Geneva.

Kuçera, Henry; Francis, W. Nelson (1967). *Computational Analysis of Present-Day American English*. Providence, Rhode Island: Brown University Press.

Mintz, Toben H. (2003). Frequent frames as a cue for grammatical categories in child directed speech. *Cognition*, 90, 91–117.

Rapp, Reinhard (2005). A practical solution to the problem of automatic part-of-speech induction from text. *Proceedings of the 43rd ACL Conference, Companion Volume,* Ann Arbor, MI, 77–80.

Rapp, Reinhard (2007). Part-of-speech discovery by clustering contextual features. In: Reinhold Decker and Hans-J. Lenz (eds.): *Advances in Data Analysis. Proceedings of the 30th Conference of the Gesellschaft für Klassifikation.* Heidelberg: Springer, 627–634.

Schütze, Hinrich (1993). Part-of-speech induction from scratch. *Proceedings of the 31st ACL Conference,* Columbus, Ohio, 251–258.

Support Vector Machines for Query-focused Summarization trained and evaluated on Pyramid data

Maria Fuentes
TALP Research Center
Universitat Politècnica de Catalunya
mfuentes@lsi.upc.edu

Enrique Alfonseca
Computer Science Departament
Universidad Autónoma de Madrid
Enrique.Alfonseca@gmail.com

Horacio Rodríguez
TALP Research Center
Universitat Politècnica de Catalunya
horacio@lsi.upc.edu

Abstract

This paper presents the use of Support Vector Machines (SVM) to detect relevant information to be included in a query-focused summary. Several SVMs are trained using information from pyramids of summary content units. Their performance is compared with the best performing systems in DUC-2005, using both ROUGE and autoPan, an automatic scoring method for pyramid evaluation.

1 Introduction

Multi-Document Summarization (MDS) is the task of condensing the most relevant information from several documents in a single one. In terms of the DUC contests[1], a query-focused summary has to provide a “brief, well-organized, fluent answer to a need for information”, described by a short query (two or three sentences). DUC participants have to synthesize 250-word sized summaries for fifty sets of 25-50 documents in answer to some queries.

In previous DUC contests, from 2001 to 2004, the manual evaluation was based on a comparison with a single human-written model. Much information in the evaluated summaries (both human and automatic) was marked as “related to the topic, but not directly expressed in the model summary”. Ideally, this relevant information should be scored during the evaluation. The pyramid method (Nenkova and Passonneau, 2004) addresses the problem by using multiple human summaries to create a gold-standard, and by exploiting the frequency of information in the human summaries in order to assign importance to different facts. However, the pyramid method requires to manually matching fragments of automatic summaries (peers) to the Semantic Content Units (SCUs) in the pyramids. AutoPan (Fuentes et al., 2005), a proposal to automate this matching process, and ROUGE are the evaluation metrics used.

As proposed by Copeck and Szpakowicz (2005), the availability of human-annotated pyramids constitutes a gold-standard that can be exploited in order to train extraction models for the summary automatic construction. This paper describes several models trained from the information in the DUC-2006 manual pyramid annotations using Support Vector Machines (SVM). The evaluation, performed on the DUC-2005 data, has allowed us to discover the best configuration for training the SVMs.

One of the first applications of supervised Machine Learning techniques in summarization was in Single-Document Summarization (Ishikawa et al., 2002). Hirao et al. (2003) used a similar approach for MDS. Fisher and Roark (2006)’s MDS system is based on perceptrons trained on previous DUC data.

2 Approach

Following the work of Hirao et al. (2003) and Kazawa et al. (2002), we propose to train SVMs for ranking the candidate sentences in order of relevance. To create the training corpus, we have used the DUC-2006 dataset, including topic descriptions, document clusters, peer and manual summaries, and pyramid evaluations as annotated during the DUC-2006 manual evaluation. From all these data, a set

[1] http://www-nlpir.nist.gov/projects/duc/

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 57–60,
Prague, June 2007.

of relevant sentences is extracted in the following way: first, the sentences in the original documents are matched with the sentences in the summaries (Copeck and Szpakowicz, 2005). Next, all document sentences that matched a summary sentence containing at least one SCU are extracted. Note that the sentences from the original documents that are not extracted in this way could either be positive (i.e. contain relevant data) or negative (i.e. irrelevant for the summary), so they are not yet labeled. Finally, an SVM is trained, as follows, on the annotated data.

Linguistic preprocessing The documents from each cluster are preprocessed using a pipe of general purpose processors performing tokenization, POS tagging, lemmatization, fine grained Named Entities (NE)s Recognition and Classification, anaphora resolution, syntactic parsing, semantic labeling (using WordNet synsets), discourse marker annotation, and semantic analysis. The same tools are used for the linguistic processing of the query. Using these data, a semantic representation of the sentence is produced, that we call *environment*. It is a semantic-network-like representation of the semantic units (nodes) and the semantic relations (edges) holding between them. This representation will be used to compute the (Fuentes et al., 2006) lexico-semantic measures between sentences.

Collection of positive instances As indicated before, every sentence from the original documents matching a summary sentence that contains at least one SCU is considered a positive example. We have used a set of features that can be classified into three groups: those extracted from the sentences, those that capture a similarity metric between the sentence and the topic description (query), and those that try to relate the cohesion between a sentence and all the other sentences in the same document or collection.

The attributes collected **from the sentences** are:

- The position of the sentence in its document.
- The number of sentences in the document.
- The number of sentences in the cluster.
- Three binary attributes indicating whether the sentence contains positive, negative and neutral discourse markers, respectively. For instance, *what's more* is positive, while *for example* and *incidentally* indicate lack of relevance.
- Two binary attributes indicating whether the sentence contains *right-directed* discourse markers (that affect the relevance of fragment after the marker, e.g. *first of all*), or discourse markers affecting both sides, e.g. *that's why*.
- Several boolean features to mark whether the sentence starts with or contains a particular word or part-of-speech tag.
- The total number of NEs included in the sentence, and the number of NEs of each kind.
- *SumBasic score* (Nenkova and Vanderwende, 2005) is originally an iterative procedure that updates word probabilities as sentences are selected for the summary. In our case, word probabilities are estimated either using only the set of words in the current document, or using all the words in the cluster.

The attributes that **depend on the query** are:

- Word-stem overlapping with the query.
- Three boolean features indicating whether the sentence contains a subject, object or indirect object dependency in common with the query.
- Overlapping between the environment predicates in the sentence and those in the query.
- Two similarity metrics calculated by expanding the query words using Google.
- *SumFocus score* (Vanderwende et al., 2006).

The **cohesion-based** attributes [2] are:

- Word-stem overlapping between this sentence and the other sentences in the same document.
- Word-stem overlapping between this sentence and the other sentences in the same cluster.
- Synset overlapping between this sentence and the other sentences in the same document.
- Synset overlapping with other sentences in the same collection.

Model training In order to train a traditional SVM, both positive and negative examples are necessary. From the pyramid data we are able to identify positive examples, but there is not enough evidence to classify the remaining sentences as positive or negative. Although One-Class Support Vector Machine (OSVM) (Manevitz and Yousef, 2001) can learn from just positive examples, according to Yu et al. (2002) they are prone to underfitting and overfitting when data is scant (which happens in

[2] The mean, median, standard deviation and histogram of the overlapping distribution are calculated and included as features.

this case), and a simple iterative procedure called Mapping-Convergence (MC) algorithm can greatly outperform OSVM (see the pseudocode in Figure 1).

```
Input: positive examples, $POS$, unlabeled examples $U$
Output: hypothesis at each iteration $h'_1, h'_2, ..., h'_k$

1. Train $h$ to identify “strong negatives” in $U$:
   $N_1$ := examples from $U$ classified as negative by $h$
   $P_1$ := examples from $U$ classified as positive by $h$
2. Set $NEG := \emptyset$ and $i := 1$
3. Loop until $N_i = \emptyset$,
   3.1. $NEG := NEG \cup N_i$
   3.2. Train $h'_i$ from $POS$ and $NEG$
   3.3. Classify $P_i$ by $h'_i$:
        $N_{i+1}$ = examples from $P_i$ classified as negative
        $P_{i+1}$ = examples from $P_i$ classified as positive
5. Return $\{h'_1, h'_2, ..., h'_k\}$
```

Figure 1: Mapping-Convergence algorithm.

The MC starts by identifying a small set of instances that are very dissimilar to the positive examples, called *strong negatives*. Next, at each iteration, a new SVM h'_i is trained using the original positive examples, and the negative examples found so far. The set of negative instances is then extended with the unlabeled instances classified as negative by h'_i.

The following settings have been tried:

- The set of positive examples has been collected either by matching document sentences to peer summary sentences (Copeck and Szpakowicz, 2005) or by matching document sentences to manual summary sentences.
- The initial set of *strong negative* examples for the MC algorithm has been either built automatically as described by Yu et al. (2002), or built by choosing manually, for each cluster, the two or three automatic summaries with lowest manual pyramid scores.
- Several SVM kernel functions have been tried.

For training, there were 6601 sentences from the original documents, out of which around 120 were negative examples and either around 100 or 500 positive examples, depending on whether the document sentences had been matched to the manual or the peer summaries. The rest were initially unlabeled.

Summary generation Given a query and a set of documents, the trained SVMs are used to rank sentences. The top ranked ones are checked to avoid redundancy using a percentage overlapping measure.

3 Evaluation Framework

The SVMs, trained on DUC-2006 data, have been tested on the DUC-2005 corpus, using the 20 clusters manually evaluated with the pyramid method. The sentence features were computed as described before. Finally, the performance of each system has been evaluated automatically using two different measures: ROUGE and autoPan.

ROUGE, the automatic procedure used in DUC, is based on n-gram co-occurrences. Both ROUGE-2 (henceforward R-2) and ROUGE-SU4 (R-SU4) has been used to rank automatic summaries.

AutoPan is a procedure for automatically matching fragments of text summaries to SCUs in pyramids, in the following way: first, the text in the SCU label and all its contributors is stemmed and stop words are removed, obtaining a set of stem vectors for each SCU. The system summary text is also stemmed and freed from stop words. Next, a search for non-overlapping windows of text which can match SCUs is carried. Each match is scored taking into account the score of the SCU as well as the number of matching stems. The solution which globally maximizes the sum of scores of all matches is found using dynamic programming techniques.

According to Fuentes et al. (2005), autoPan scores are highly correlated to the manual pyramid scores. Furthermore, autoPan also correlates well with manual responsiveness and both ROUGE metrics.[3]

3.1 Results

Positive	Strong neg.	R-2	R-SU4	autoPan
peer	pyramid scores	**0.071**	**0.131**	**0.072**
	(Yu et al., 2002)	0.036	0.089	0.024
manual	pyramid scores	0.025	0.075	0.024
	(Yu et al., 2002)	0.018	0.063	0.009

Table 1: ROUGE and autoPan results using different SVMs.

Table 1 shows the results obtained, from which some trends can be found: firstly, the SVMs trained using the set of positive examples obtained from peer summaries consistently outperform SVMs trained using the examples obtained from the manual summaries. This may be due to the fact that the

[3] In DUC-2005 pyramids were created using 7 manual summaries, while in DUC-2006 only 4 were used. For that reason, better correlations are obtained in DUC-2005 data.

number of positive examples is much higher in the first case (on average 48,9 vs. 12,75 examples per cluster). Secondly, generating automatically a set with seed negative examples for the M-C algorithm, as indicated by Yu et al. (2002), usually performs worse than choosing the strong negative examples from the SCU annotation. This may be due to the fact that its quality is better, even though the amount of seed negative examples is one order of magnitude smaller in this case (11.9 examples in average). Finally, the best results are obtained when using a RBF kernel, while previous summarization work (Hirao et al., 2003) uses polynomial kernels.

The proposed system attains an autoPan value of 0.072, while the best DUC-2005 one (Daumé III and Marcu, 2005) obtains an autoPan of 0.081. The difference is not statistically significant. (Daumé III and Marcu, 2005) system also scored highest in responsiveness (manually evaluated at NIST).

However, concerning ROUGE measures, the best participant (Ye et al., 2005) has an R-2 score of 0.078 (confidence interval [0.073–0.080]) and an R-SU4 score of 0.139 [0.135–0.142], when evaluated on the 20 clusters used here. The proposed system again is comparable to the best system in DUC-2005 in terms of responsiveness, Daumé III and Marcu (2005)'s R-2 score was 0.071 [0.067–0.074] and R-SU4 was 0.126 [0.123–0.129] and it is better than the DUC-2005 Fisher and Roark supervised approach with an R-2 of 0.066 and an R-SU4 of 0.122.

4 Conclusions and future work

The pyramid annotations are a valuable source of information for training automatically text summarization systems using Machine Learning techniques. We explore different possibilities for applying them in training SVMs to rank sentences in order of relevance to the query. Structural, cohesion-based and query-dependent features are used for training.

The experiments have provided some insights on which can be the best way to exploit the annotations. Obtaining the positive examples from the annotations of the peer summaries is probably better because most of the peer systems are extract-based, while the manual ones are abstract-based. Also, using a very small set of strong negative example seeds seems to perform better than choosing them automatically with Yu et al. (2002)'s procedure.

In the future we plan to include features from adjacent sentences (Fisher and Roark, 2006) and use rouge scores to initially select negative examples.

Acknowledgments

Work partially funded by the CHIL project, IST-2004506969.

References

T. Copeck and S. Szpakowicz. 2005. Leveraging pyramids. In *Proc. DUC-2005*, Vancouver, Canada.

Hal Daumé III and Daniel Marcu. 2005. Bayesian summarization at DUC and a suggestion for extrinsic evaluation. In *Proc. DUC-2005*, Vancouver, Canada.

S. Fisher and B. Roark. 2006. Query-focused summarization by supervised sentence ranking and skewed word distributions. In *Proc. DUC-2006*, New York, USA.

M. Fuentes, E. Gonzàlez, D. Ferrés, and H. Rodríguez. 2005. QASUM-TALP at DUC 2005 automatically evaluated with the pyramid based metric autopan. In *Proc. DUC-2005*.

M. Fuentes, H. Rodríguez, J. Turmo, and D. Ferrés. 2006. FEMsum at DUC 2006: Semantic-based approach integrated in a flexible eclectic multitask summarizer architecture. In *Proc. DUC-2006*, New York, USA.

T. Hirao, J. Suzuki, H. Isozaki, and E. Maeda. 2003. Ntt's multiple document summarization system for DUC2003. In *Proc. DUC-2003*.

K. Ishikawa, S. Ando, S. Doi, and A. Okumura. 2002. Trainable automatic text summarization using segmentation of sentence. In *Proc. 2002 NTCIR 3 TSC workshop*.

H. Kazawa, T. Hirao, and E. Maeda. 2002. Ranking SVM and its application to sentence selection. In *Proc. 2002 Workshop on Information-Based Induction Science (IBIS-2002)*.

L.M. Manevitz and M. Yousef. 2001. One-class SVM for document classification. *Journal of Machine Learning Research*.

A. Nenkova and R. Passonneau. 2004. Evaluating content selection in summarization: The pyramid method. In *Proc. HLT/NAACL 2004*, Boston, USA.

A. Nenkova and L. Vanderwende. 2005. The impact of frequency on summarization. Technical Report MSR-TR-2005-101, Microsoft Research.

L. Vanderwende, H. Suzuki, and C. Brockett. 2006. Microsoft research at DUC 2006: Task-focused summarization with sentence simplification and lexical expansion. In *Proc. DUC-2006*, New York, USA.

S. Ye, L. Qiu, and T.S. Chua. 2005. NUS at DUC 2005: Understanding documents via concept links. In *Proc. DUC-2005*.

H. Yu, J. Han, and K. C-C. Chang. 2002. PEBL: Positive example-based learning for web page classification using SVM. In *Proc. ACM SIGKDD International Conference on Knowledge Discovery in Databases (KDD02)*, New York.

A Joint Statistical Model for Simultaneous Word Spacing and Spelling Error Correction for Korean

Hyungjong Noh* Jeong-Won Cha** Gary Geunbae Lee*

***Department of Computer Science and Engineering**
Pohang University of Science & Technology (POSTECH)
San 31, Hyoja-Dong, Pohang, 790-784, Republic of Korea

**** Changwon National University**
Department of Computer information & Communication
9 Sarim-dong, Changwon Gyeongnam, Korea 641-773

nohhj@postech.ac.kr jcha@changwon.ac.kr gblee@postech.ac.kr

Abstract

This paper presents noisy-channel based Korean preprocessor system, which corrects word spacing and typographical errors. The proposed algorithm corrects both errors simultaneously. Using Eojeol transition pattern dictionary and statistical data such as Eumjeol n-gram and Jaso transition probabilities, the algorithm minimizes the usage of huge word dictionaries.

1 Introduction

With increasing usages of messenger and SMS, we need an efficient text normalizer that processes colloquial style sentences. As in the case of general literary sentences, correcting word spacing error and spelling error is the very essential problem with colloquial style sentences.

In order to correct word spacing errors, many algorithms were used, which can be divided into statistical algorithms and rule-based algorithms. Statistical algorithms generally use character n-gram (Eojeol[1] or Eumjeol[2] n-gram in Korean) (Kang and Woo, 2001; Kwon, 2002) or noisy-channel model (Gao et. al., 2003). Rule-based algorithms are mostly heuristic algorithms that reflect linguistic knowledge (Yang et al., 2005) to solve word spacing problem. Word spacing problem is treated especially in Japanese or Chinese, which does not use word boundary, or Korean, which is normally segmented into Eojeols, not into words or morphemes.

The previous algorithms for spelling error correction basically use a word dictionary. Each word in a sentence is compared to word dictionary entries, and if the word is not in the dictionary, then the system assumes that the word has spelling errors. Then corrected candidate words are suggested by the system from the word dictionary, according to some metric to measure the similarity between the target word and its candidate word, such as edit-distance (Kashyap and Oommen, 1984; Mays et al., 1991).

But these previous algorithms have a critical limitation: They all corrected word spacing errors and spelling errors separately. Word spacing algorithms define the problem as a task for determining whether to insert the delimiter between characters or not. Since the determination is made according to the characters, the algorithms cannot work if the characters have spelling errors. Likewise, algorithms for solving spelling error problem cannot work well with word spacing errors.

To cope with the limitation, there is an algorithm proposed for Japanese (Nagata, 1996). Japanese sentence cannot be divided into words, but into chunks (bunsetsu in Japanese), like Eojeol in Korean. The proposed system is for sentences recognized by OCR, and it uses character transition probabilities and POS (part of speech) tag n-gram. However it needs a word dictionary and takes long time for searching many character combinations.

[1] Eojeol is a Korean spacing unit which consists of one or more Eumjeols (morphemes).

[2] Eumjeol is a Korean syllable.

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 61–64,
Prague, June 2007. ©2007 Association for Computational Linguistics

We propose a new algorithm which can correct both word spacing error and spelling error simultaneously for Korean. This algorithm is based on noisy-channel model, which uses Jaso[3] transition probabilities and Eojeol transition probabilities to create spelling correction candidates. Candidates are increased in number by inserting the blank characters on the created candidates, which cover the spacing error correction candidates. We find the best candidate sentence from the networks of Jaso/Eojeol candidates. This method decreases the size of Eojeol transition pattern dictionary and corrects the patterns which are not in the dictionary.

The remainder of this paper is as follows: Section 2 describes why we use Jaso transition probability for Korean. Section 3 describes the proposed model in detail. Section 4 provides the experiment results and analyses. Finally, section 5 presents our conclusion.

2 Spelling Error Correction with Jaso Transition[4] Probabilities

We can use Eumjeol transition probabilities or Jaso transition probabilities for spelling error correction for Korean. We choose Jaso transition probabilities because there are several advantages. Since an Eumjeol is a combination of 3 Jasos, the number of all possible Eumjeols is much larger than that of all possible Jasos. In other words, Jaso-based language model is smaller than Eumjeol-based language model. Various errors in Eumjeol (even if they do not appear as an Eumjeol pattern in a training corpus) can be corrected by correction in Jaso unit. Also, Jaso transition probabilities can be extracted from relatively small corpus. This merit is very important since we do not normally have such a huge corpus which is very hard to collect, since we have to pair the spelling errors with corresponding corrections.

We obtain probabilities differently for each case: single Jaso transition case, two Jaso's transition case, and more than two Jasos transition case.

In single Jaso transition case, the spelling errors are corrected by only one Jaso transition (e.g. 같애요→같아요 / ㅐ→ㅏ). The case of correcting by deleting Jaso is also one of the single Jaso transition case (나와욧→나와요 / ㅅ→X[5]). The Jaso transition probabilities are calculated by counting the transition frequencies in a training corpus.

In two Jaso's transition case, the spelling errors are corrected by adjacent two Jasos transition (춥오→초보 / ㅂㅇ→X ㅂ). In this case, we treat two Jaso's as one transition unit. The transition probability calculation is the same as above.

In more than two Jaso's transition case, the spelling errors cannot be corrected only by Jaso transition (걍→그냥). In this case, we treat the whole Eojeols as one transition unit, and build an Eojeol transition pattern dictionary for these special cases.

3 A Joint Statistical Model for Word Spacing and Spelling Error Correction

3.1 Problem Definition

Given a sentence T which includes both word spacing errors and spelling errors, we create correction candidates C from T, and find the best candidate C' that has the highest transition probability from C.

$$C' = \arg\max_C P(C \mid T). \qquad (1)$$

3.2 Model Description

A given sentence T and candidates C consist of Eumjeol s_i and the blank character b_i.

$$T = s_1 b_1 s_2 b_2 s_3 b_3 \ldots s_n b_n .$$
$$C = s_1 b_1 s_2 b_2 s_3 b_3 \ldots s_n b_n . \qquad (2)$$

(n is the number of Eumjeols)

Eumjeol s_i consists of 3 Jasos, Choseong (onset), Jungseong (nucleus), and Jongseong (coda). The empty Jaso is defined as 'X'. b_i is 'B' when the blank exists, and 'Φ' when the blank does not exist.

$$s_i = j_{i1} j_{i2} j_{i3} . \qquad (3)$$

(j_{i1} : Choseong, j_{i2} : Jungseong, j_{i3} : Jongseong)

Now we apply Bayes' Rule for C':

$$\begin{aligned} C' &= \arg\max_C P(C \mid T) \\ &= \arg\max_C P(T \mid C) P(C) / P(T) \\ &= \arg\max_C P(T \mid C) P(C). \end{aligned} \qquad (4)$$

[3] Jaso is a Korean character.

[4] 'Transition' means the correct character is changed to other character due to some causes, such as typographical errors.

[5] 'X' indicates that there is no Jaso in that position.

$P(C)$ can be obtained using trigrams of Eumjeols (with the blank character) that C includes.

$$P(C)=\prod_{i=1}^{n}P(c_i \mid c_{i-1}c_{i-2}),\ c=s \text{ or } b. \quad (5)$$

And $P(T \mid C)$ can be written as multiplication of each Jaso transition probability and the blank character transition probability.

$$P(T \mid C)=\prod_{i=1}^{n}P(s_i \mid s_i^{'})$$

$$=\prod_{i=1}^{n}[P(j_{i1} \mid j_{i1}^{'})P(j_{i2} \mid j_{i2}^{'})P(j_{i3} \mid j_{i3}^{'})P(b_i \mid b_i^{'})]. \quad (6)$$

We use logarithm of $P(C \mid T)$ in implementation. Figure 1 shows how the system creates the Jaso candidates network.

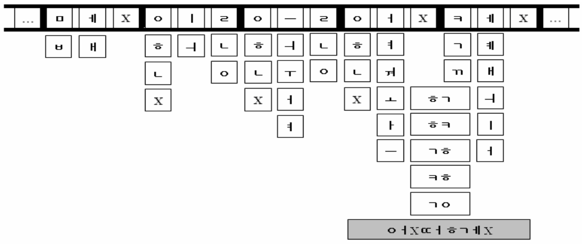


Figure 1: An example[6] of Jaso candidate network.

In Figure 1, the topmost line is the sequence of Jasos of the input sentence. Each Eumjeol in the sentence is decomposed into 3 Jasos as above, and each Jaso has its own correction candidates. For example, Jaso ‘ㅇ’ at 4th column has its candidates ‘ㅎ’, ‘ㄴ’ and ‘X’. And two jaso’s ‘Xㅋ’ at 13th and 14th column has its candidates ‘ㅎㄱ’, ‘ㅎㅋ’, ’ㄱㅎ’, ’ㅋㅎ’, and ‘ㄱㅇ’. The undermost gray square is an Eojeol (which is decomposed into Jasos) candidate ‘ㅇㅓXㄸㅓㅎㄱㅔX’ created from ‘ㅇㅓXㅋㅔX’. Each jaso candidate has its own transition probability, $\log P(j_{ik} \mid j_{ik}^{'})$[7], that is used for calculating $P(C \mid T)$.

In order to calculate $P(C)$, we need Eumjeol-based candidate network. Hence, we convert the above Jaso candidate network into Eumjeol/Eojeol candidate network. Figure 2 shows part of the final network briefly. At this time, the blank characters ‘B’ and ‘Φ’ are inserted into each Eumjeol/Eojeol candidates. To find the best path from the candidates, we conduct viterbi-search from leftmost node corresponding to the beginning of the sentence. When Eumjeol/Eojeol candidates are selected, the algorithm prunes the candidates according to the accumulated probabilities, doing beam search. Once the best path is found, the sentence corrected by both spacing and spelling errors is extracted by backtracking the path. In Figure 2, thick squares represent the nodes selected by the best path.

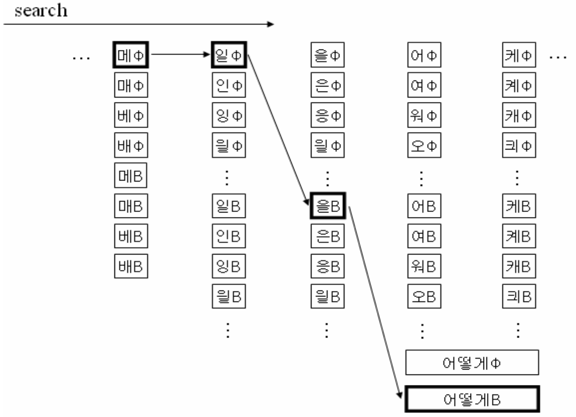

search
...
메Φ 매Φ 베Φ 배Φ 메B 매B 베B 배B
일Φ 인Φ 잉Φ 윌Φ 일B 인B 잉B 윌B
을Φ 은Φ 응Φ 윌Φ 을B 은B 응B 윌B
어Φ 여Φ 워Φ 오Φ 어B 여B 워B 오B
케Φ 켸Φ 캐Φ 킈Φ 케B 켸B 캐B 킈B
...
어떻게Φ
어떻게B

Figure 2: A final Eumjeol/Eojeol candidate network[8]

4 Experiments and Analyses

4.1 Corpus Information

	Training	Test
Sentences	60076	6006
Eojeols	302397	30376
Error Sentences (%)	15335 (25.53)	1512 (25.17)
Error Eojeols (%)	31297 (10.35)	3111 (10.24)

Table 1: Corpus information

Table 1 shows the information of corpus which is used for experiments. All corpora are obtained from Korean web chatting site log. Each corpus has pair of sentences, sentences containing errors and sentences with those errors corrected. Jaso transition patterns and Eojeol transition patterns are extracted from training corpus. Also, Eumjeol n-grams are also obtained as a language model.

[6] The example sentence is “데체메일을어케보내는거지”.

[7] In real implementation, we used “a*log$P(j_{ik}|j'_{ik})$ + b” by determining constants a and b with parameter optimization (a = 1.0, b = 3.0).

[8] The final corrected sentence is “대체 메일을 어떻게 보내는 거지”.

4.2 Experiment Results and Analyses

We used two separate Eumjeol n-grams as language models for experiments. N-gram A is obtained from only training corpus and n-gram B is obtained from all training and test corpora. All accuracies are measured based on Eojeol unit.

Table 2 shows the results of word spacing error correction only for the test corpus.

	n-gram A	n-gram B
Accuracy	91.03%	96.00%

Table 2: The word spacing error correction results

The results of both word spacing error and spelling error correction are shown in Table 3. Error containing test corpus (the blank characters are all deleted) was applied to this evaluation.

System	n-gram A	n-gram B
Basic joint model	88.34%	93.83%

Table 3: The joint model results

Table 4 shows the results of the same experiment, without deleting the blank characters in the test corpus. The experiment shows that our joint model has a flexibility of utilizing already existing blanks (spacing) in the input sentence.

System	n-gram A	n-gram B
Baseline	89.35%	89.35%
Basic joint model with keeping the blank characters	90.35%	95.25%

Table 4: The joint model results without deleting the exist spaces

As shown above, the performance is dependent of the language model (n-gram) performance. Jaso transition probabilities can be obtained easily from small corpus because the number of Jaso is very small, under 100, in contrast with Eumjeol.

Using the existing blank information is also an important factor. If test sentences have no or few blank characters, then we simply use joint algorithm to correct both errors. But when the test sentences already have some blank characters, we can use the information since some of the spacing can be given by the user. By keeping the blank characters, we can get better accuracy because blank insertion errors are generally fewer than the blank deletion errors in the corpus.

5 Conclusions

We proposed a joint text preprocessing model that can correct both word spacing and spelling errors simultaneously for Korean. To our best knowledge, this is the first model which can handle inter-related errors between spacing and spelling in Korean. The usage and size of the word dictionaries are decreased by using Jaso statistical probabilities effectively.

6 Acknowledgement

This work was supported in part by MIC & IITA through IT Leading R&D Support Project.

References

Jianfeng Gao, Mu Li and Chang-Ning Huang. 2003. *Improved Source-Channel Models for Chinese Word Segmentation*. Proceedings of the 41st Annual Meeting of the ACL, pp. 272-279

Seung-Shik Kang and Chong-Woo Woo. 2001. *Automatic Segmentation of Words Using Syllable Bigram Statistics*. Proceedings of 6th Natural Language Processing Pacific Rim Symposium, pp. 729-732

R. L Kashyap, B. J. Oommen. 1984. *Spelling Correction Using Probabilistic Methods*. Pattern Recognition Letters, pp. 147-154

Oh-Wook Kwon. 2002. *Korean Word Segmentation and Compound-noun Decomposition Using Markov Chain and Syllable N-gram*. The Journal of the Acoustical Society of Korea, pp. 274-283.

Mu Li, Muhua Zhu, Yang Zhang and Ming Zhou. 2006. *Exploring Distributional Similarity Based Models for Query Spelling Correction*. Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL, pp. 1025-1032

Eric Mays, Fred J. Damerau and Robert L. Mercer. 1991. *Context Based Spelling Correction*. IP&M, pp. 517-522.

Masaaki Nagata. 1996. *Context-Based Spelling Correction for Japanese OCR*. Proceedings of the 16th conference on Computational Linguistics, pp. 806-811

Christoper C. Yang and K. W. Li. 2005. *A Heuristic Method Based on a Statistical Approach for Chinese Text Segmentation*. Journal of the American Society for Information Science and Technology, pp. 1438-1447.

An Approximate Approach for Training Polynomial Kernel SVMs in Linear Time

Yu-Chieh Wu
Dept. of Computer Science and
Information Engineering
National Central University
Taoyuan, Taiwan
bcbb@db.csie.ncu.edu.tw

Jie-Chi Yang
Graduate Institute of Network Learning Technology
National Central University
Taoyuan, Taiwan
yang@cl.ncu.edu.tw

Yue-Shi Lee
Dept. of Computer Science and
Information Engineering
Ming Chuan University
Taoyuan, Taiwan
lees@mcu.edu.tw

Abstract

Kernel methods such as support vector machines (SVMs) have attracted a great deal of popularity in the machine learning and natural language processing (NLP) communities. Polynomial kernel SVMs showed very competitive accuracy in many NLP problems, like part-of-speech tagging and chunking. However, these methods are usually too inefficient to be applied to large dataset and real time purpose. In this paper, we propose an approximate method to analogy polynomial kernel with efficient data mining approaches. To prevent exponential-scaled testing time complexity, we also present a new method for speeding up SVM classifying which does independent to the polynomial degree d. The experimental results showed that our method is 16.94 and 450 times faster than traditional polynomial kernel in terms of training and testing respectively.

1 Introduction

Kernel methods, for example support vector machines (SVM) (Vapnik, 1995) are successfully applied to many natural language processing (NLP) problems. They yielded very competitive and satisfactory performance in many classification tasks, such as part-of-speech (POS) tagging (Gimenez and Marquez, 2003), shallow parsing (Kudo and Matsumoto, 2001, 2004; Lee and Wu, 2007), named entity recognition (Isozaki and Kazawa, 2002), and parsing (Nivre et al., 2006).

In particular, the use of polynomial kernel SVM implicitly takes the feature combinations into account instead of explicitly combines features. By setting with polynomial kernel degree (i.e., *d*), different number of feature conjunctions can be implicitly computed. In this way, polynomial kernel SVM is often better than linear kernel which did not use feature conjunctions. However, the training and testing time costs for polynomial kernel SVM is far slow than the linear kernel. For example, it took one day to train the CoNLL-2000 task with polynomial kernel SVM, while the testing speed is merely 20-30 words per second (Kudo and Matsumoto, 2001). Although the author provided the solution for fast classifying with polynomial kernel (Kudo and Matsumoto, 2004), the training time is still inefficient. Nevertheless, the testing time of their method exponentially scales with polynomial kernel degree *d*, i.e., $O(|X|^d)$ where $|X|$ denotes as the length of example X.

On the contrary, even the linear kernel SVM simply disregards the effect of feature combinations during training and testing, it performs not only more efficient than polynomial kernel, but also can be improved through directly appending features derived from the set of feature combinations. Examples include bigram, trigram, etc. Nevertheless, selecting the feature conjunctions was manually and heuristically encoded and should perform amount of validation trials to discover which is useful or not. In recent years, several studies had reported that the training time of linear kernel SVM can be reduced to linear time (Joachims, 2006; Keerthi and DeCoste, 2005). But they did not and difficult to be extent to polynomial kernels.

In this paper, we propose an approximate approach to extend the linear kernel SVM toward polynomial. By introducing the well-known sequential pattern mining approach (Pei et al., 2004),

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 65–68,
Prague, June 2007. ©2007 Association for Computational Linguistics

frequent feature conjunctions, namely patterns could be discovered and also kept as expand feature space. We then adopt the mined patterns to re-represent the training/testing examples. Subsequently, we use the off-the-shelf linear kernel SVM algorithm to perform training and testing. Besides, to exponential-scaled testing time complexity, we propose a new classification method for speeding up the SVM testing. Rather than enumerating all patterns for each example, our method requires O(F_{avg}*N_{avg}) which is independent to the polynomial kernel degree. F_{avg} is the average number of frequent features per example, while the N_{avg} is the average number of patterns per feature.

2 SVM and Kernel Methods

Suppose we have the training instance set for binary classification problem:

$$(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n),\ x_i \in \Re^D,\ y_i \in \{+1, -1\}$$

where x_i is a feature vector in D-dimension space of the i-th example, and y_i is the label of xi either positive or negative. The training of SVMs involves in minimize the following object (primal form, soft-margin) (Vapnik, 1995):

$$\text{minimize}: W(\alpha) = \frac{1}{2}\overrightarrow{W} \cdot \overrightarrow{W} + C\sum_{i=1}^{n} Loss(\overrightarrow{W}x_i, y_i) \qquad (1)$$

The loss function indicates the loss of training error. Usually, the hinge-loss is used (Keerthi and DeCoste, 2005). The factor C in (1) is a parameter that allows one to trade off training error and margin. A small value for C will increase the number of training errors.

To determine the class (+1 or -1) of an example x can be judged by computing the following equation.

$$y(x) = \text{sign}((\sum_{x_i \in SVs} \alpha_i y_i K(x, x_i)) + b) \qquad (2)$$

α_i is the weight of training example x_i (α_i>0), and b denotes as a threshold. Here the xi should be the support vectors (SVs), and are representative of training examples. The kernel function K is the kernel mapping function, which might map from $\Re^D$ to $\Re^{D'}$ (usually D<<D'). The natural linear kernel simply uses the dot-product as (3).

$$K(x, x_i) = dot(x, x_i) \qquad (3)$$

A polynomial kernel of degree d is given by (4).

$$K(x, x_i) = (1 + dot(x, x_i))^d \qquad (4)$$

One can design or employ off-the-shelf kernel types for particular applications. In particular to the use of polynomial kernel-based SVM, it was shown to be the most successful kernels for many natural language processing (NLP) problems (Kudo and Matsumoto, 2001; Isozaki and Kazawa, 2002; Nivre et al., 2006).

It is known that the dot-product (linear form) represents the most efficient kernel computing which can produce the output value by linearly combining all support vectors such as

$$y(x) = \text{sign}(dot(x, w) + b) \qquad \text{where } w = \sum_{x_i \in SVs} \alpha_i y_i x_i \qquad (5)$$

By combining (2) and (4), the determination of an example of x using the polynomial kernel can be shown as follows.

$$y(x) = \text{sign}((\sum_{x_i \in SVs} \alpha_i y_i (dot(x, x_i) + 1)^d) + b) \qquad (6)$$

Usually, degree d is set more than 1. When d is set as 1, the polynomial kernel backs-off to linear kernel. Although the effectiveness of polynomial kernel, it can not be shown to linearly combine all support vectors into one weight vector whereas it requires computing the kernel function (4) for each support vector x_i. The situation is even worse when the number of support vectors become huge (Kudo and Matsumoto, 2004). Therefore, whether in training or testing phrase, the cost of kernel computations is far more expensive than linear kernel.

3 Approximate Polynomial Kernel

In 2004, Kudo and Matsumoto (2004) derived both implicitly (6) and explicitly form of polynomial kernel. They indicated that the use of explicitly enumerate the feature combinations is equivalent to the polynomial kernel (see Lemma 1 and Example 1, Kudo and Matsumoto, 2004) which shared the same view of (Cumby and Roth, 2003).

We follow the similar idea of the above studies that requires explicitly enumerated all feature combinations. To meet with our problem, we employ the well-known sequential pattern mining algorithm, namely PrefixSpan (Pei et al., 2004) to efficient mine the frequent patterns. However, directly adopt the algorithm is not a good idea. To fit with SVM, we modify the original PrefixSpan algorithm according to the following constraints.

Given a set features, the PrefixSpan mines the frequent patterns which occurs more than predefined minimum support in the training set and limited in the length of predefined d, which is equivalent to the polynomial kernel degree d. For example, if the minimum support is 5, and d=2, then a feature combination (f_i, f_j) must appear more than 5 times in set of x.

Definition 1 (*Frequent single-item sequence*): Given a set of feature vectors x, minimum support, and d, mining the frequent patterns (feature combinations) is to mine the patterns in the single-item sequence database.

Lemma 2 (*Ordered feature vector*): For each example, the feature vector could be transformed into an ordered item (feature) list, i.e., $f_1<f_2<\ldots<f_{max}$ where f_{max} is the highest dimension of the example.

Proof. It is very easy to sort an unordered feature vector into the ordered list with conventional sorting algorithm.

Definition 3 (*Uniqueness of the features per example*): Given the set of mined patterns, for any feature f_i, it is impossible to appear more than once in the same pattern.

Different from conventional sequential pattern mining method, in feature combination mining for SVM only contains a set of feature vectors each of which is independently treated. In other words, no compound features in the vector. If it exists, one can simply expand the compound features as another new feature.

By means of the above constraints, mining the frequent patterns can be reduced to mining the limited length of frequent patterns in the single-item database (set of ordered vectors). Furthermore, during each phase, we need only focus on finding the "frequent single features" to expand previous phase. More detail implementation issues can refer (Pei et al., 2004).

3.1 Speed-up Testing

To efficiently expand new features for the original feature vectors, we propose a new method to fast discovery patterns. Essentially, the PrefixSpan algorithm gradually expands one item from previous result which can be viewed as a tree growing. An example can be found in Figure 1.

Each node in Figure 1 is the associate feature of root. The whole patterns expanded by f_j can be represented as the path from root to each node. For example, pattern (f_j, f_k, f_m, f_r) can be found via traversing the tree starting from f_j. In this way, we can re-expand the original feature vector via visiting corresponding trees for each feature.

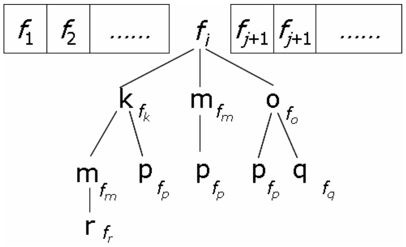


Figure 1: The tree representation of feature f_j

Table 1: Encoding frequent patterns with DFS array representation

Level	0	1	2	3	2	1	2	1	2	2
Label	Root	k	m	r	p	m	p	o	p	q
Item	f_j	f_k	f_m	f_r	f_p	f_m	f_p	f_o	f_p	f_q

However, traversing arrays is much more efficient than visiting trees. Therefore, we adopt the l^2-sequences encoding method based on the DFS (depth-first-search) sequence as (Wang et al., 2004) to represent the trees. An l^2-sequence does not only store the label information but also take the node level into account. Examples can be found in Table 1.

Theorem 4 (*Uniqueness of l^2-sequence*): Given trees T_1, and T_2, their l^2-sequences are identical if and only if T_1 and T_2 are isomorphic, i.e., there exists a one-to-one mapping for set of nodes, node labels, edges, and root nodes.

Proof. see theorem 1 in (Wang et al., 2004).

Definition 5 (*Ascend-descend relation*): Given a node k of feature f_k in l^2-sequence, all of the descendant of k that rooted by k have the greater feature numbers than f_k.

Definition 6 (*Limited visiting space*): Given the highest feature $f_{\max}$ of vector X, and f_k rooted l^2-sequence, if $f_{\max}<f_k$, then we can not find any pattern that prefix by f_k.

Both definitions 5 and 6 strictly follow lemma 2 that kept the ordered relations among features. For example, once node k could be found in X, it is unnecessary to visit its children. More specifically, to determine whether a frequent pattern is in X, we need to compare feature vector of X and l^2-sequence database. It is clearly that the time complexity of our method is O(F_{avg}*N_{avg}) where F_{avg} is the average number of frequent features per example, while the N_{avg} is the average length of l^2-sequence. In other words, our method does not dependent on the polynomial kernel degree.

4 Experiments

To evaluate our method, we examine the well-known shallow parsing task which is the task of CoNLL-2000[1]. We also adopted the released perl-evaluator to measure the recall/precision/f1 rates. The used feature consists of word, POS, orthographic, affix(2-4 prefix/suffix letters), and previous chunk tags in the two words context window size (the same as (Lee and Wu, 2007)). We limited the features should at least appear more than twice in the training set.

For the learning algorithm, we replicate the modified finite Newton SVM as learner which can be trained in linear time (Keerthi and DeCoste, 2005). We also compare our method with the standard linear and polynomial kernels with SVM^{light}[2].

4.1 Results

Table 2 lists the experimental results on the CoNLL-2000 shallow parsing task. Table 3 compares the testing speed of different feature expansion techniques, namely, array visiting (our method) and enumeration.

Table 2: Experimental results for CoNLL-2000 shallow parsing task

CoNLL-2000	F1	Mining Time	Training Time	Testing Time
Linear Kernel	93.15	N/A	0.53hr	2.57s
Polynomial(*d*=2)	94.19	N/A	11.52hr	3189.62s
Polynomial(*d*=3)	93.95	N/A	19.43hr	6539.75s
Our Method (*d*=2,sup=0.01)	93.71	<10s	0.68hr	6.54s
Our Method (*d*=3,sup=0.01)	93.46	<15s	0.79hr	9.95s

Table 3: Classification time performance of enumeration and array visiting techniques

CoNLL-2000	Array visiting		Enumeration	
	d=2	*d*=3	*d*=2	*d*=3
Testing time	6.54s	9.95s	4.79s	11.73s
Chunking speed (words/sec)	7244.19	4761.50	9890.81	4038.95

It is not surprising that the best performance was obtained by the classical polynomial kernel. But the limitation is that the slow in training and testing time costs. The most efficient method is linear kernel SVM but it does not as accurate as polynomial kernel. However, our method stands for both efficiency and accuracy in this experiment. In terms of training time, it slightly slower than the linear kernel, while it is 16.94 and ~450 times faster than polynomial kernel in training and testing. Besides, the pattern mining time is far smaller than SVM training.

As listed in Table 3, we can see that our method provide a more efficient solution to feature expansion when *d* is set more than two. Also it demonstrates that when *d* is small, the enumerate-based method is a better choice (see PKE in (Kudo and Matsumoto, 2004)).

5 Conclusion

This paper presents an approximate method for extending linear kernel SVM to analogy polynomial-like computing. The advantage of this method is that it does not require maintaining the cost of support vectors in training, while achieves satisfactory result. On the other hand, we also propose a new method for speeding up classification which is independent to the polynomial kernel degree. The experimental results showed that our method close to the performance of polynomial kernel SVM and better than the linear kernel. In terms of efficiency, our method did not only improve 16.94 times faster in training and 450 times in testing, but also faster than previous similar studies.

References

Chad Cumby and Dan Roth. 2003. Kernel methods for relational learning. International Conference on Machine Learning, pages 104-114.

Hideki Isozaki and Hideto Kazawa. 2002. *Efficient support vector classifiers for named entity recognition*. International Conference on Computational Linguistics, pages 1-7.

Jian Pei, Jiawei Han, Behzad Mortazavi-Asl, Jianyong Wang, Helen Pinto, Qiming Chen, Umeshwar Dayal and Mei-Chun Hsu. 2004. Mining Sequential Patterns by Pattern-Growth: The Prefix Span Approach. IEEE Trans. on Knowledge and Data Engineering, 16(11): 1424-1440.

Sathiya Keerthi and Dennis DeCoste. 2005. *A modified finite Newton method for fast solution of large scale linear SVMs*. Journal of Machine Learning Research. 6: 341-361.

Taku Kudo and Yuji Matsumoto. 2001. *Fast methods for kernel-based text analysis*. Annual Meeting of the Association for Computational Linguistics, pages 24-31.

Taku Kudo and Yuji Matsumoto. 2001. *Chunking with support vector machines*. Annual Meetings of the North American Chapter and the Association for the Computational Linguistics.

Yue-Shi Lee and Yu-Chieh Wu. 2007. *A Robust Multilingual Portable Phrase Chunking System*. Expert Systems with Applications, 33(3): 1-26.

Vladimir N. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer.

Chen Wang, Mingsheng Hong, Jian Pei, Haofeng Zhou, Wei Wang and Baile Shi. 2004. *Efficient Pattern-Growth Methods for Frequent Tree Pattern Mining*. Pacific knowledge discovery in database (PAKDD).

[1] http://www.cnts.ua.ac.be/conll2000/chunking/

[2] http://svmlight.joachims.org/

Rethinking Chinese Word Segmentation: Tokenization, Character Classification, or Wordbreak Identification

Chu-Ren Huang
Institute of Linguistics
Academia Sinica,Taiwan
churen@gate.sinica.edu.tw

Petr Šimon
Institute of Linguistics
Academia Sinica,Taiwan
sim@klubko.net

Shu-Kai Hsieh
DoFLAL
NIU, Taiwan
shukai@gmail.com

Laurent Prévot
CLLE-ERSS, CNRS
Université de Toulouse, France
prevot@univ-tlse2.fr

Abstract

This paper addresses two remaining challenges in Chinese word segmentation. The challenge in HLT is to find a robust segmentation method that requires no prior lexical knowledge and no extensive training to adapt to new types of data. The challenge in modelling human cognition and acquisition it to segment words efficiently without using knowledge of wordhood. We propose a radical method of word segmentation to meet both challenges. The most critical concept that we introduce is that Chinese word segmentation is the classification of a string of character-boundaries (CB's) into either word-boundaries (WB's) and non-word-boundaries. In Chinese, CB's are delimited and distributed in between two characters. Hence we can use the distributional properties of CB among the background character strings to predict which CB's are WB's.

1 Introduction: modeling and theoretical challenges

The fact that word segmentation remains a main research topic in the field of Chinese language processing indicates that there maybe unresolved theoretical and processing issues. In terms of processing, the fact is that none of exiting algorithms is robust enough to reliably segment unfamiliar types of texts before fine-tuning with massive training data. It is true that performance of participating teams have steadily improved since the first SigHAN Chinese segmentation bakeoff (Sproat and Emerson, 2004). Bakeoff 3 in 2006 produced best f-scores at 95% and higher. However, these can only be achieved after training with the pre-segmented training dataset. This is still very far away from real-world application where any varieties of Chinese texts must be successfully segmented without prior training for HLT applications.

In terms of modeling, all exiting algorithms suffer from the same dilemma. Word segmentation is supposed to identify word boundaries in a running text, and words defined by these boundaries are then compared with the mental/electronic lexicon for POS tagging and meaning assignments. All existing segmentation algorithms, however, presuppose and/or utilize a large lexical databases (e.g. (Chen and Liu, 1992) and many subsequent works), or uses the position of characters in a word as the basis for segmentation (Xue, 2003).

In terms of processing model, this is a contradiction since segmentation should be the pre-requisite of dictionary lookup and should not presuppose lexical information. In terms of cognitive modeling, such as for acquisition, the model must be able to account for how words can be successfully segmented and learned by a child/speaker without formal training or a priori knowledge of that word. All current models assume comprehensive lexical knowledge.

2 Previous work

Tokenization model. The classical model, described in (Chen and Liu, 1992) and still adopted in many recent works, considers text segmentation as a

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 69–72,
Prague, June 2007. ©2007 Association for Computational Linguistics

tokenization. Segmentation is typically divided into two stages: dictionary lookup and out of vocabulary (OOV) word identification. This approach requires comparing and matching tens of thousands of dictionary entries in addition to guessing thousands of OOV words. That is, this is a 10^4x10^4 scale mapping problem with unavoidable data sparseness.

More precisely the task consist in finding all sequences of characters $C_i, \ldots, C_n$ such that $[C_i, \ldots C_n]$ either matches an entry in the lexicon or is guessed to be so by an unknown word resolution algorithm. One typical kind of the complexity this model faces is the overlapping ambiguity where e.g. a string $[Ci-1, Ci, Ci+1]$ contains multiple substrings, such as $[Ci-1, Ci,]$ and $[Ci, Ci+1]$, which are entries in the dictionary. The degree of such ambiguities is estimated to fall between 5% to 20% (Chiang et al., 1996; Meng and Ip, 1999).

2.1 Character classification model

A popular recent innovation addresses the scale and sparseness problem by modeling segmentation as character classification (Xue, 2003; Gao et al., 2004). This approach observes that by classifying characters as word-initial, word-final, penultimate, etc., word segmentation can be reduced to a simple classification problem which involves about 6,000 characters and around 10 positional classes. Hence the complexity is reduced and the data sparseness problem resolved. It is not surprising then that the character classification approach consistently yields better results than the tokenization approach. This approach, however, still leaves two fundamental questions unanswered. In terms of modeling, using character classification to predict segmentation not only increases the complexity but also necessarily creates a lower ceiling of performance In terms of language use, actual distribution of characters is affected by various factors involving linguistic variation, such as topic, genre, region, etc. Hence the robustness of the character classification approach is restricted.

The character classification model typically classifies all characters present in a string into at least three classes: word Initial, Middle or Final positions, with possible additional classification for word-middle characters. Word boundaries are inferred based on the character classes of 'Initial' or 'Final'.

This method typically yields better result than the tokenization model. For instance, Huang and Zhao (2006) claims to have a f-score of around 97% for various SIGHAN bakeoff tasks.

3 A radical model

We propose a radical model that returns to the core issue of word segmentation in Chinese. Crucially, we no longer pre-suppose any lexical knowledge. Any unsegmented text is viewed as a string of character-breaks (CB's) which are evenly distributed and delimited by characters. The characters are not considered as components of words, instead, they are contextual background providing information about the likelihood of whether each CB is also a wordbreak (WB). In other words, we model Chinese word segmentation as wordbreak (WB) identification which takes all CB's as candidates and returns a subset which also serves as wordbreaks. More crucially, this model can be trained efficiently with a small corpus marked with wordbreaks and does not require any lexical database.

3.1 General idea

Any Chinese text is envisioned as sequence of characters and character-boundaries $CB_0C1CB_1C_2 \ldots CB_{i-1}C_iCB_i \ldots CB_{n-1}C_nCB_n$ The segmentation task is reduced to finding all CBs which are also wordbreaks WB.

3.2 Modeling character-based information

Since CBs are all the same and do not carry any information, we have to rely on their distribution among different characters to obtain useful information for modeling. In a segmented corpus, each WB can be differentiated from a non-WB CB by the character string before and after it. We can assume a reduced model where either one character immediately before and after a CB is considered or two characters (bigram). These options correspond to consider (i) only word-initial and word-final positions (hereafter the 2-CB-model or 2CBM) or (ii) to add second and penultimate positions (hereafter the 4-CB-model or 4CBM). All these positions are well-attested as morphologically significant.

3.3 The nature of segmentation

It is important to note that in this approaches, although characters are recognized, unlike (Xue, 2003) and Huang et al. (2006), charactes simply are in the background. That is, they are the necessary delimiter, which allows us to look at the string of CB's and obtaining distributional information of them.

4 Implementation and experiments

In this section we slightly change our notation to allow for more precise explanation. As noted before, Chinese text can be formalized as a sequence of characters and intervals as illustrated in we call this representation an *interval form.*

$c_1 I_1 c_2 I_2 \ldots c_{n-1} I_{n-1} c_n$.

In such a representation, each interval I_k is either classified as a plain character boundary (CB) or as a word boundary (WB).

We represent the neighborhood of the character c_i as $(c_{i-2}, I_{i-2}, c_{i-1}, I_{i-1}, c_i, I_i, c_{i+1}, I_{i+1})$, which we can be simplified as $(I_{-2}, I_{-1}, c_i, I_{+1}, I_{+2})$ by removing all the neighboring characters and retaining only the intervals.

4.1 Data collection models

This section makes use of the notation introduced above for presenting several models accounting for character-interval class co-occurrence.

Word based model. In this model, statistical data about word boundary frequencies for each character is retrieved word-wise. For example, in the case of a monosyllabic word only two word boundaries are considered: one before and one after the character that constitutes the monosyllabic word in question.

The method consists in mapping all the Chinese characters available in the training corpus to a vector of word boundary frequencies. These frequencies are normalized by the total frequency of the character in a corpus and thus represent probability of a word boundary occurring at a specified position with regard to the character.

Let us consider for example, a tri-syllabic word $W = c_1 c_2 c_3$, that can be rewritten as the following interval form as $W^I = I^B_{-1} c_1 I^N_1 c_2 I^N_2 c_3 I^B_3$.

In this interval form, each interval I_k is marked as word boundary ^{B} or ^{N} for intervals within words. When we consider a particular character c_1 in W, there is a word boundary at index -1 and 3. We store this information in a mapping $c_1 = \{-1 : 1, 3 : 1\}$. For each occurrence of this character in the corpus, we modify the character vector accordingly, each WB corresponding to an increment of the relevant position in the vector. Every character in every word of the corpus in processed in a similar way.

Obviously, each character yields only information about positions of word boundaries of a word this particular character belongs to. This means that the index I_{-1} and I_3 are not necessarily incremented everytime (e.g. for monosyllabic and bi-syllabic words)

Sliding window model. This model does not operate on words, but within a window of a give size $(span)$ sliding through the corpus. We have experimented this method with a window of size 4. Let us consider a string, $s =$ "$c_1 c_2 c_3 c_4$" which is not necessarily a word and is rewritten into an interval form as $s^I =$ "$c_1 I_1 c_2 I_2 c_3 I_3 c_4 I_4$". We store the co-occurrence character/word boundaries information in a fixed size $(span)$ vector.

For example, we collect the information for character c_3 and thus arrive at a vector $c_3 = (I_1, I_2, I_3, I_4)$, where 1 is incremented at the respective position $if I_k = WB$, zero otherwise.

This model provides slightly different information that the previous one. For example, if a sequence of four characters is segmented as $c_1 I^N_1 c_2 I^B_2 c_3 I^B_3 c_4 I^B_4$ (a sequence of one bi-syllabic and two monosyllabic words), for c_3 we would also get probability of I_4, i.e. an interval with index +2 . In other words, this model enables to learn WB probability across words.

4.2 Training corpus

In the next step, we convert our training corpus into a corpus of interval vectors of specified dimension. Let's assume we are using dimension $span = 4$. Each value in such a vector represents the probability of this interval to be a word boundary. This probability is assigned by character for each position with regard to the interval. For example, we have segmented corpus $C = c_1 I_1 c_2 I_2 \ldots c_{n-1} I_{n-1} c_n$, where each I_k is labeled as B for word boundary or N for non-boundary.

In the second step, we move our 4-sized window through the corpus and for each interval we query a character at the corresponding position from the interval to retrieve the word boundary occurrence probability. This procedure provides us with a vector of 4 probability values for each interval. Since we are creating this training corpus from an already segmented text, a class (B or N) is assigned to each interval.

The testing corpus (unsegmented) is encoded in a similar way, but does not contain the class labels B and N.

Finally, we automatically assign probability of 0.5 for unseen events.

4.3 Predicting word boundary with a classifier

The Sinica corpus contains 6820 types of characters (including Chinese characters, numbers, punctuation, Latin alphabet, etc.). When the Sinica corpus is converted into our interval vector corpus, it provides 14.4 million labeled interval vectors. In this first study we have implement a baseline model, without any pre-processing of punctuation, numbers, names.

A decision tree classifier (Ruggieri, 2004) has been adopted to overcome the non-linearity issue. The classifier was trained on the whole Sinica corpus, i.e. on 14.4 million interval vectors. Due to space limit, actual bakeoff experiment result will be reported in our poster presentation.

Our best results is based on the sliding window model, which provides better results. It has to be emphasized that the test corpora were not processed in any way, i.e. our method is sufficiently robust to account for a large number of ambiguities like numerals, foreign words.

5 Conclusion

In this paper, we presented a radical and robust model of Chinese segmentation which is supported by initial experiment results. The model does not pre-suppose any lexical information and it treats character strings as context which provides information on the possible classification of character-breaks as word-breaks. We are confident that once a standard model of pre-segmentation, using textual encoding information to identify WB's which involves non-Chinese characters, will enable us to achieve even better results. In addition, we are looking at other alternative formalisms and tools to implement this model to achieve the optimal results. Other possible extensions including experiments to simulate acquisition of wordhood knowledge to provide support of cognitive modeling, similar to the simulation work on categorization in Chinese by (Redington et al., 1995). Last, but not the least, we will explore the possibility of implementing a sharable tool for robust segmentation for all Chinese texts without training.

References

Academia Sinica Balanced Corpus of Modern Chinese. http://www.sinica.edu.tw/SinicaCorpus/

Chen K.J and Liu S.H. 1992. *Word Identification for Mandarin Chinese sentences*. Proceedings of the 14th conference on Computational Linguistics, p.101-107, France.

Chiang,T.-H., J.-S. Chang, M.-Y. Lin and K.-Y. Su. 1996. *Statistical Word Segmentation*. In C.-R. Huang, K.-J. Chen and B.K. T'sou (eds.): Journal of Chinese Linguistics, Monograph Series, Number 9, Readings in Chinese Natural Language Processing, pp. 147-173.

Gao, J. and A. Wu and Mu Li and C.-N.Huang and H. Li and X. Xia and H. Qin. 2004. *Adaptive Chinese Word Segmentation*. In Proceedings of ACL-2004.

Meng, H. and C. W. Ip. 1999. *An Analytical Study of Transformational Tagging for Chinese Text*. In. Proceedings of ROCLING XII. 101-122. Taipei

Ruggieri S. 2004. *YaDT: Yet another Decision Tree builder*. Proceedings of the 16th International Conference on Tools with Artificial Intelligence (ICTAI 2004): 260-265. IEEE Press, November 2004.

Richard Sproat and Thomas Emerson. 2003. *The First International Chinese Word Segmentation Bakeoff*. Proceedings of the Second SIGHAN Workshop on Chinese Language Processing, Sapporo, Japan, July 2003.

Xue, N. 2003. *Chinese Word Segmentation as Character Tagging*. Computational Linguistics and Chinese Language Processing. 8(1): 29-48

Redington, M. and N. Chater and C. Huang and L. Chang and K. Chen. 1995. *The Universality of Simple Distributional Methods: Identifying Syntactic Categories in Mandarin Chinese*. Presented at the Proceedings of the International Conference on Cognitive Science and Natural Language Processing. Dublin City University.

A Feature Based Approach to Leveraging Context for Classifying Newsgroup Style Discussion Segments

Yi-Chia Wang, Mahesh Joshi
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213
{yichiaw,maheshj}@cs.cmu.edu

Carolyn Penstein Rosé
Language Technologies Institute/
Human-Computer Interaction Institute
Carnegie Mellon University
Pittsburgh, PA 15213
cprose@cs.cmu.edu

Abstract

On a multi-dimensional text categorization task, we compare the effectiveness of a feature based approach with the use of a state-of-the-art sequential learning technique that has proven successful for tasks such as "email act classification". Our evaluation demonstrates for the three separate dimensions of a well established annotation scheme that novel thread based features have a greater and more consistent impact on classification performance.

1 Introduction

The problem of information overload in personal communication media such as email, instant messaging, and on-line discussion boards is a well documented phenomenon (Bellotti, 2005). Because of this, conversation summarization is an area with a great potential impact (Zechner, 2001). What is strikingly different about this form of summarization from summarization of expository text is that the summary may include more than just the content, such as the style and structure of the conversation (Roman et al., 2006). In this paper we focus on a classification task that will eventually be used to enable this form of conversation summarization by providing indicators of the quality of group functioning and argumentation.

Lacson and colleagues (2006) describe a form of conversation summarization where a classification approach is first applied to segments of a conversation in order to identify regions of the conversation related to different types of information. This aids in structuring a useful summary. In this paper, we describe work in progress towards a different form of conversation summarization that similarly leverages a text classification approach. We focus on newsgroup style interactions. The goal of assessing the quality of interactions in that context is to enable the quality and nature of discussions that occur within an on-line discussion board to be communicated in a summary to a potential newcomer or group moderators.

We propose to adopt an approach developed in the computer supported collaborative learning (CSCL) community for measuring the quality of interactions in a threaded, online discussion forum using a multi-dimensional annotation scheme (Weinberger & Fischer, 2006). Using this annotation scheme, messages are segmented into idea units and then coded with several independent dimensions, three of which are relevant for our work, namely micro-argumentation, macro-argumentation, and social modes of co-construction, which categorizes spans of text as belonging to one of five consensus building categories. By coding segments with this annotation scheme, it is possible to measure the extent to which group members' arguments are well formed or the extent to which they are engaging in functional or dysfunctional consensus building behavior.

This work can be seen as analogous to work on "email act classification" (Carvalho & Cohen, 2005). However, while in some ways the structure of newsgroup style interaction is more straightforward than email based interaction because of the unambiguous thread structure (Carvalho & Cohen, 2005), what makes this particularly challenging

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 73–76,
Prague, June 2007.

from a technical standpoint is that the structure of this type of conversation is multi-leveled, as we describe in greater depth below.

We investigate the use of state-of-the-art sequential learning techniques that have proven successful for email act classification in comparison with a feature based approach. Our evaluation demonstrates for the three separate dimensions of a context oriented annotation scheme that novel thread based features have a greater and more consistent impact on classification performance.

2 Data and Coding

We make use of an available annotated corpus of discussion data where groups of three students discuss case studies in an on-line, newsgroup style discussion environment (Weinberger & Fischer, 2006). This corpus is structurally more complex than the data sets used previously to demonstrate the advantages of using sequential learning techniques for identifying email acts (Carvalho & Cohen, 2005). In the email act corpus, each message as a whole is assigned one or more codes. Thus, the history of a span of text is defined in terms of the thread structure of an email conversation. However, in the Weinberger and Fischer corpus, each message is segmented into idea units. Thus, a span of text has a context within a message, defined by the sequence of text spans within that message, as well as a context from the larger thread structure.

The Weinberger and Fischer annotation scheme has seven dimensions, three of which are relevant for our work.

1. *Micro-level of argumentation* [4 categories] How an individual argument consists of a claim which can be supported by a ground with warrant and/or specified by a qualifier
2. *Macro-level of argumentation* [6 categories] Argumentation sequences are examined in terms of how learners connect individual arguments to create a more complex argument (for example, consisting of an argument, a counter-argument, and integration)
3. *Social Modes of Co-Construction* [6 categories] To what degree or in what ways learners refer to the contributions of their learning partners, including externalizations, elicitations, quick consensus building, integration oriented consensus building, or conflict oriented consensus building, or other.

For the two argumentation dimensions, the most natural application of sequential learning techniques is by defining the history of a span of text in terms of the sequence of spans of text within a message, since although arguments may build on previous messages, there is also a structure to the argument within a single message. For the Social Modes of Co-construction dimension, it is less clear. However, we have experimented with both ways of defining the history and have not observed any benefit of sequential learning techniques by defining the history for sequential learning in terms of previous messages. Thus, for all three dimensions, we report results for histories defined within a single message in our evaluation below.

3 Feature Based Approach

In previous text classification research, more attention to the selection of predictive features has been done for text classification problems where very subtle distinctions must be made or where the size of spans of text being classified is relatively small. Both of these are true of our work. For the base features, we began with typical text features extracted from the raw text, including unstemmed unigrams and punctuation. We did not remove stop words, although we did remove features that occured less than 5 times in the corpus. We also included a feature that indicated the number of words in the segment.

Thread Structure Features. The simplest context-oriented feature we can add based on the threaded structure is a number indicating the depth in the thread where a message appears. We refer to this feature as *deep.* This is expected to improve performance to the extent that thread initial messages may be rhetorically distinct from messages that occur further down in the thread. The other context oriented feature related to the thread structure is derived from relationships between spans of text appearing in the parent and child messages. This feature is meant to indicate how semantically related a span of text is to the spans of text in the parent message. This is computed using the minimum of all cosine distance measures between the vector representation of the span of text and that of each of the spans of text in all parent messages, which is a typical shallow measure of semantic similarity. The smallest such distance measure is included as a feature indicating how related the current span of text is to a parent message.

Sequence-Oriented Features. We hypothesized that the sequence of codes within a message follows a semi-regular structure. In particular, the discussion environment used to collect the Weinberger and Fischer corpus inserts prompts into the message buffers before messages are composed in order to structure the interaction. Users fill in text underneath these prompts. Sometimes they quote material from a previous message before inserting their own comments. We hypothesized that whether or not a piece of quoted material appears before a span of text might influence which code is appropriate. Thus, we constructed the *fsm* feature, which indicates the state of a simple finite-state automaton that only has two states. The automaton is set to initial state (q_0) at the top of a message. It makes a transition to state (q_1) when it encounters a quoted span of text. Once in state (q_1), the automaton remains in this state until it encounters a prompt. On encountering a prompt it makes a transition back to the initial state (q_0). The purpose is to indicate places where users are likely to make a comment in reference to something another participant in the conversation has already contributed.

4 Evaluation

The purpose of our evaluation is to contrast our proposed feature based approach with a state-of-the-art sequential learning technique (Collins, 2002). Both approaches are designed to leverage context for the purpose of increasing classification accuracy on a classification task where the codes refer to the role a span of text plays in context.

We evaluate these two approaches alone and in combination over the same data but with three different sets of codes, namely the three relevant dimensions of the Weinberger and Fischer annotation scheme. In all cases, we employ a 10-fold cross-validation methodology, where we apply a feature selection wrapper in such as way as to select the 100 best features over the training set on each fold, and then to apply this feature space and the trained model to the test set. The complete corpus comprises about 250 discussions of the participants. From this we have run our experiments with a subset of this data, using altogether 1250 annotated text segments. Trained coders categorized each segment using this multi-dimensional annotation scheme, in each case achieving a level of agreement exceeding .7 Kappa both for segmentation and coding of all dimensions as previously published (Weinberger & Fischer, 2006).

For each dimension, we first evaluate alternative combinations of features using SMO, Weka's implementation of Support Vector Machines (Witten & Frank, 2005). For a sequential learning algorithm, we make use of the Collins Perceptron Learner (Collins, 2002). When using the Collins Perceptron Learner, in all cases we evaluate combinations of alternative history sizes (0 and 1) and alternative feature sets (base and base+AllContext). In our experimentation we have evaluated larger history sizes as well, but the performance was consistently worse as the history size grew larger than 1. Thus, we only report results for history sizes of 0 and 1.

Our evaluation demonstrates that we achieve a much greater impact on performance with carefully designed, automatically extractable context oriented features. In all cases we are able to achieve a statistically significant improvement by adding context oriented features, and only achieve a statistically significant improvement using sequential learning for one dimension, and only in the absence of context oriented features.

4.1 Feature Based Approach

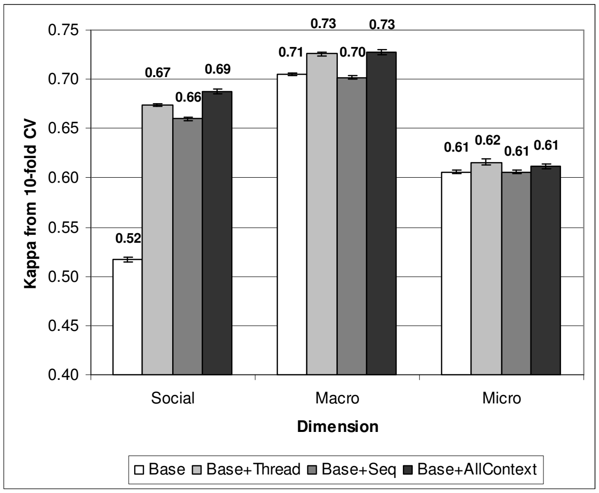


Figure 1. Results with alternative features sets

We first evaluated the feature based approach across all three dimensions and demonstrate that statistically significant improvements are achieved on all dimensions by adding context oriented features. The most dramatic results are achieved on the Social Modes of Co-Construction dimension (See Figure 1). All pairwise contrasts between alternative feature sets within this dimension are statistically significant. In the other dimensions, while Base+Thread is a significant improvement over Base, there is no significant difference between Base+Thread and Base+AllContext.

4.2 Sequential Learning

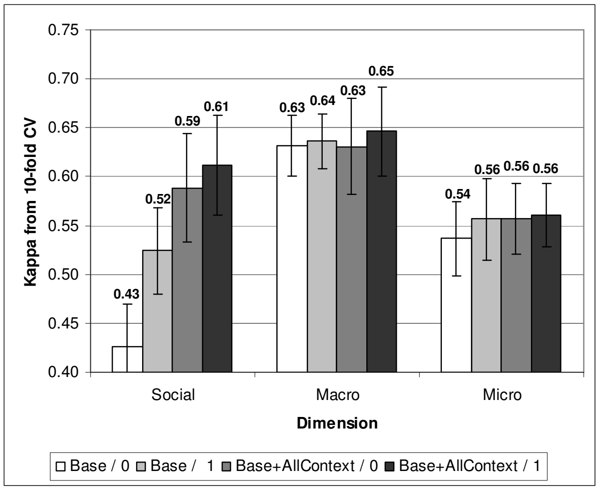


Figure 2. Results with Sequential Learning

The results for sequential learning are weaker than for the feature based (See Figure 2). While the Collins Perceptron learner possesses the capability of modeling sequential dependencies between codes, which SMO does not possess, it is not necessarily a more powerful learner. On this data set, the Collins Perceptron learner consistently performs worse that SMO. Even restricting our evaluation of sequential learning to a comparison between the Collins Perceptron learner with a history of 0 (i.e., no history) with the same learner using a history of 1, we only see a statistically significant improvement on the Social Modes of Co-Construction dimension. This is when only using base features, although the trend was consistently in favor of a history of 1 over 0. Note that the standard deviation in the performance across folds was much higher with the Collins Perceptron learner, so that a much greater difference in average would be required in order to achieve statistical significance. Performance over a validation set was always worse with larger history sizes than 1.

5 Conclusions

We have described work towards an approach to conversation summarization where an assessment of conversational quality along multiple process dimensions is reported. We make use of a well-established annotation scheme developed in the CSCL community. Our evaluation demonstrates that thread based features have a greater and more consistent impact on performance with this data.

This work was supported by the National Science Foundation grant number SBE0354420, and Office of Naval Research, Cognitive and Neural Sciences Division Grant N00014-05-1-0043.

References

Bellotti, V., Ducheneaut, N., Howard, M. Smith, I., Grinter, R. (2005). Quality versus Quantity: Email-centric task management and its relation with overload. Human-Computer Interaction, 2005, vol. 20

Carvalho, V. & Cohen, W. (2005). On the Collective Classification of Email “Speech Acts”, Proceedings of SIGIR ‘2005.

Collins, M (2002). Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proceedings of EMNLP 2002*.

Lacson, R., Barzilay, R., & Long, W. (2006). Automatic analysis of medical dialogue in the homehemodialysis domain: structure induction and summarization, *Journal of Biomedical Informatics* 39(5), pp541-555.

Roman, N., Piwek, P., & Carvalho, A. (2006). Politeness and Bias in Dialogue Summarization : Two Exploratory Studies, in J. Shanahan, Y. Qu, & J. Wiebe (Eds.) *Computing Attitude and Affect in Text: Theory and Applications, the Information Retrieval Series.*

Weinberger, A., & Fischer, F. (2006). A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Computers & Education*, 46, 71-95.

Witten, I. H. & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*, second edition, Elsevier: San Francisco.

Zechner, K. (2001). Automatic Generation of Concise Summaries of Spoken Dialogues in Unrestricted Domains. *Proceedings of ACM SIG-IR 2001.*

Ensemble Document Clustering
Using Weighted Hypergraph Generated by NMF

Hiroyuki Shinnou, Minoru Sasaki
Ibaraki University,
4-12-1 Nakanarusawa, Hitachi,
Ibaraki, Japan 316-8511
{shinnou,msasaki}@mx.ibaraki.ac.jp

Abstract

In this paper, we propose a new ensemble document clustering method. The novelty of our method is the use of Non-negative Matrix Factorization (NMF) in the generation phase and a weighted hypergraph in the integration phase. In our experiment, we compared our method with some clustering methods. Our method achieved the best results.

1 Introduction

In this paper, we propose a new ensemble document clustering method using Non-negative Matrix Factorization (NMF) in the generation phase and a weighted hypergraph in the integration phase.

Document clustering is the task of dividing a document's data set into groups based on document similarity. This is the basic intelligent procedure, and is important in text mining systems (M. W. Berry, 2003). As the specific application, relevant feedback in IR, where retrieved documents are clustered, is actively researched (Hearst and Pedersen, 1996)(Kummamuru et al., 2004).

In document clustering, the document is represented as a vector, which typically uses the *"bag of word"* model and the TF-IDF term weight. A vector represented in this manner is highly dimensional and sparse. Thus, in document clustering, a dimensional reduction method such as PCA or SVD is applied before actual clustering (Boley et al., 1999)(Deerwester et al., 1990). Dimensional reduction maps data in a high-dimensional space into a low-dimensional space, and improves both clustering accuracy and speed.

NMF is a dimensional reduction method (Xu et al., 2003) that is based on the "aspect model" used in the Probabilistic Latent Semantic Indexing (Hofmann, 1999). Because the axis in the reduced space by NMF corresponds to a topic, the reduced vector represents the clustering result. For a given term-document matrix and cluster number, we can obtain the NMF result with an iterative procedure (Lee and Seung, 2000). However, this iteration does not always converge to a global optimum solution. That is, NMF results depend on the initial value. The standard countermeasure for this problem is to generate multiple clustering results by changing the initial value, and then select the best clustering result estimated by an object function. However, this selection often fails because the object function does not always measure clustering accuracy.

To overcome this problem, we use ensemble clustering, which combines multiple clustering results to obtain an accurate clustering result.

Ensemble clustering consists of generation and integration phases. The generation phase produces multiple clustering results. Many strategies have been proposed to achieve this goal, including random initialization (Fred and Jain, 2002), feature extraction based on random projection (Fern and Brodley, 2003) and the combination of sets of "weak" partitions (Topchy et al., 2003). The integration phase, as the name implies, integrates multiple clustering results to improve the accuracy of the final clustering result. This phase primarily relies on two methods. The first method constructs a new simi-

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 77–80,
Prague, June 2007. ©2007 Association for Computational Linguistics

larity matrix from multiple clustering results (Fred and Jain, 2002). The second method constructs new vectors for each instance data using multiple clustering results (Strehl and Ghosh, 2002). Both methods apply the clustering procedure to the new object to obtain the final clustering result.

Our method generates multiple clustering results by random initialization of the NMF, and integrates them with a weighted hypergraph instead of the standard hypergraph (Strehl and Ghosh, 2002). An advantage of our method is that the weighted hypergraph can be directly obtained from the NMF result.

In our experiment, we compared the k-means, NMF, the ensemble method using a standard hypergraph and the ensemble method using a weighted hypergraph. Our method achieved the best results.

2 NMF

The NMF decomposes the $m \times n$ term-document matrix X to the $m \times k$ matrix U and the transposed matrix of the $n \times k$ matrix V (Xu et al., 2003), where k is the number of clusters; that is,

$$X = UV^T.$$

The i-th document d_i corresponds to the i-th row vector of V; that is, $d_i = (v_{i1}, v_{i2}, \cdots, v_{ik})$. The cluster number is obtained from $\arg\max_{j \in 1:k} v_{ij}$.

For a given term-document matrix X, we can obtain U and V by the following iteration (Lee and Seung, 2000):

$$u_{ij} \leftarrow u_{ij} \frac{(XV)_{ij}}{(UV^TV)_{ij}} \quad (1)$$

$$v_{ij} \leftarrow v_{ij} \frac{(X^TU)_{ij}}{(VU^TU)_{ij}}. \quad (2)$$

Here, u_{ij}, v_{ij} and $(X)_{ij}$ represent the i-th row and the j-th column element of U, V and X respectively.

After each iteration, U must be normalized as follows:

$$u_{ij} \leftarrow \frac{u_{ij}}{\sqrt{\sum_i u_{ij}^2}}. \quad (3)$$

Either the fixed maximum iteration number, or the distance J between X and UV^T stops the iteration:

$$J = ||X - UV^T||_F. \quad (4)$$

In NMF, the clustering result depends on the initial values. Generally, we conduct NMF several times with random initialization, and then select the clustering result with the smallest value of Eq.4. The value of Eq.4 represents the NMF decomposition error and not the clustering error. Thus, we cannot alway select the best result.

3 Ensemble clustering

3.1 Hypergraph data representation

To overcome the above mentioned problem, we used ensemble clustering. Ensemble clustering consists of generation and integration phases. The first phase generates multiple clustering results with random initialization of the NMF. We integrated them with the hypergraph proposed in (Strehl and Ghosh, 2002).

Suppose that the generation phase produces m clustering results, and each result has k clusters. In this case, the dimension of the new vector is km. The $(k(i-1)+c)$-th dimensional value of the data d is defined as follows: If the c-th cluster of the i-th clustering result includes the data d, the value is 1. Otherwise, the value is 0. Thus, the km dimensional vector for the data d is constructed.

Consider a simple example, where $k = 3$, $m = 4$ and the data set is $\{d_1, d_2, \cdots, d_7\}$. We generate four clustering results. Supposing that the first clustering result is $\{d_1, d_2, d_5\}, \{d_3, d_4\}, \{d_6, d_7\}$, we can obtain the 1st, 2nd and 3rd column of the hypergraph as follows:

$$\begin{array}{cc} \begin{array}{c} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \end{array} & \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \end{array}.$$

Repeating the procedure produces a total of four matrices from four clustering results. Connecting these four partial matrices, we obtain the following 7×12 matrix, which is the hypergraph.

$$\begin{array}{cc} \begin{array}{c} d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \end{array} & \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix} \end{array}$$

3.2 Weighted hypergraph vs. standard hypergraph

Each element of the hypergraph is 0 or 1. However, the element value must be real because it represents the membership degree for the corresponding cluster.

Fortunately, the matrix V produced by NMF describes the membership degree. Thus, we assign the real value described in V to the element of the hypergraph whose value is 1. Figure 1 shows an example of this procedure. Our method uses this weighted hypergraph, instead of a standard hypergraph for integration.

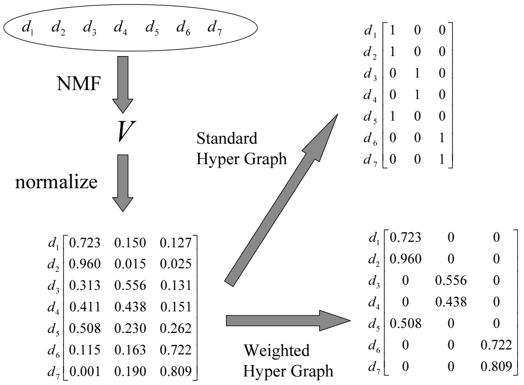


Figure 1: Weighted hypergraph through the matrix V

4 Experiment

To confirm the effectiveness of our method, we compared the k-means, NMF, the ensemble method using a standard hypergraph and the ensemble method using a weighted hypergraph.

In our experiment, we use 18 document data sets provided at `http://glaros.dtc.umn.edu/gkhome/cluto/cluto/download`.

The document vector is not normalized for each data set. We normalize them using TF-IDF.

Table 1 shows the result of the experiment [1]. The value in the table represents entropy, and the smaller it is, the better the clustering result.

In NMF, we generated 20 clustering results using random initialization, and selected the clustering result with the smallest decomposition error. The selected clustering result is shown as "NMF" in Table 1. "NMF means" in Table 1 is the average of 20 entropy values for 20 clustering results. The "standard hypergraph" and "weighted hypergraph" in Table 1 show the results of the ensemble method obtained using the two hypergraph types. Table 1 shows the effectiveness of our method.

5 Related works

When we generate multiple clustering results, the number of clusters in each clustering is fixed to the number of clusters in the final clustering result. This is not a limitation of our ensemble method. Any number is available for each clustering. Experience shows that the ensemble clustering using k-means succeeds when each clustering has many clusters, and they are combined into fewer clusters, which is a heuristics that has been reported (Fred and Jain, 2002), and is available for our method

Our method uses the weighted hypergraph, which is constructed by changing the value 1 in the standard hypergraph to the corresponding real value in the matrix V. Taking this idea one step further, it may be good to change the value 0 in the standard hypergraph to its real value. In this case, the weighted hypergraph is constructed by only connecting multiple Vs. We tested this complete weighted hypergraph, and the results are shown as "hypergraph V" in Table 1.

"Hypergraph V" was better than the standard hypergraph, but worse than our method. Furthermore, the value 0 may be useful because we can use the graph spectrum clustering method (Ding et al., 2001), which is a powerful clustering method for the spare hypergraph.

In clustering, the cluster label is unassigned. However, if cluster labeling is possible, we can use many techniques in the ensemble learning (Breiman, 1996). Cluster labeling is not difficult when there are two or three clusters. We plan to study this approach of the labeling cluster first and then using the techniques from ensemble learning.

6 Conclusion

This paper proposed a new ensemble document clustering method. The novelty of our method is the use

[1] We used the clustering toolkit CLUTO for clustering the hypergraph.

Table 1: Document data sets and Experiment results

Data	# of	# of	# of	k-means	NMF	NMF	Standard	Weighted	Hypergraph
	doc.	terms	classes			means	hypergraph	hypergraph	V
cacmcisi	4663	41681	2	0.750	0.817	0.693	0.691	**0.690**	0.778
cranmed	2431	41681	2	**0.113**	0.963	0.792	0.750	0.450	0.525
fbis	2463	2000	17	0.610	0.393	0.406	0.408	**0.381**	0.402
hitech	2301	126373	6	**0.585**	0.679	0.705	0.683	0.684	0.688
k1a	2340	21839	20	0.374	0.393	0.377	0.386	**0.351**	0.366
k1b	2340	21839	6	0.221	0.259	0.238	0.456	0.216	**0.205**
la1	3204	31472	6	0.641	0.464	0.515	**0.458**	0.459	0.491
la2	3075	31472	6	0.620	0.576	0.551	0.548	**0.468**	0.486
re0	1504	2886	13	**0.368**	0.419	0.401	0.383	0.379	0.378
re1	1657	3758	25	0.374	0.364	0.346	0.334	**0.325**	0.337
reviews	4069	126373	5	**0.364**	0.398	0.538	0.416	0.408	0.391
tr11	414	6429	9	0.349	0.338	0.311	0.300	0.304	**0.280**
tr12	313	5804	8	0.493	0.332	0.375	0.308	**0.307**	0.316
tr23	204	5832	6	0.527	0.485	0.489	0.493	0.521	**0.474**
tr31	927	10128	7	0.385	0.402	0.383	0.343	0.334	**0.310**
tr41	878	7454	10	0.277	0.358	0.299	**0.245**	0.270	0.340
tr45	690	8261	10	0.397	0.345	0.328	0.277	**0.274**	0.380
wap	1560	6460	20	0.408	0.371	0.374	0.336	**0.327**	0.344
Average	1946.2	27874.5	9.9	0.436	0.464	0.451	0.434	**0.397**	0.416

of NMF in the generation phase and a weighted hypergraph in the integration phase. One advantage of our method is that the weighted hypergraph can be obtained directly from the NMF results. Our experiment showed the effectiveness of our method using 18 document data sets. In the future, we will use an ensemble learning technique by labeling clusters.

References

D. Boley, M. L. Gini, R. Gross, E. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. 1999. Document categorization and query generation on the world wide web using webace. *Artificial Intelligence Review*, 13(5-6):365–391.

L. Breiman. 1996. Bagging predictors. *Machine Learning*, 24(2):123–140.

S. C. Deerwester, S. T. Dumais, T. K. Landauer, G. W. Furnas, and R. A. Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, 41(6):391–407.

C. Ding, X. He, H. Zha, M. Gu, and H. Simon. 2001. Spectral Min-max Cut for Graph Partitioning and Data Clustering. In *Lawrence Berkeley National Lab. Tech. report 47848*.

X. Z. Fern and C. E. Brodley. 2003. Random Projection for High Dimensional Data Clustering: A Cluster Ensemble Approach. In *the 20th International Conference of Machine Learning (ICML-03)*.

A.L.N. Fred and A. K. Jain. 2002. Data Clustering Using Evidence Accumulation. In *the 16th international conference on pattern recognition*, pages 276–280.

M. A. Hearst and J. O. Pedersen. 1996. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proceedings of SIGIR-96*, pages 76–84.

T. Hofmann. 1999. Probabilistic Latent Semantic Indexing. In *Proceedings of the 22nd Annual ACM Conference on Research and Development in Information Retrieval*, pages 50–57.

K. Kummamuru, R. Lotlikar, S. Roy, K. Singal, and R. Krishnapuram. 2004. A Hierarchical Monothetic Document Clustering Algorithm for Summarization and Browsing Search Results. In *Proceedings of WWW-04*, pages 658–665.

D. D. Lee and H. S. Seung. 2000. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562.

M. W. Berry, editor. 2003. *Survey of Text Mining: Clustering, Classification, and Retrieval*. Springer.

A. Strehl and J. Ghosh. 2002. Cluster Ensembles - A Knowledge Reuse Framework for Combining Multiple Partitions. In *Conference on Artificial Intelligence (AAAI-2002)*, pages 93–98.

A. Topchy, A. K. Jain, and W. Punch. 2003. Combining Multiple Weak Clusterings.

W. Xu, X. Liu, and Y. Gong. 2003. Document clustering based on non-negative matrix factorization. In *Proceedings of SIGIR-03*, pages 267–273.

Using Error-Correcting Output Codes with Model-Refinement to Boost Centroid Text Classifier

Songbo Tan
Information Security Center, ICT, P.O. Box 2704, Beijing, 100080, China
tansongbo@software.ict.ac.cn, tansongbo@gmail.com

Abstract

In this work, we investigate the use of error-correcting output codes (ECOC) for boosting centroid text classifier. The implementation framework is to decompose one multi-class problem into multiple binary problems and then learn the individual binary classification problems by centroid classifier. However, this kind of decomposition incurs considerable bias for centroid classifier, which results in noticeable degradation of performance for centroid classifier. In order to address this issue, we use Model-Refinement to adjust this so-called bias. The basic idea is to take advantage of misclassified examples in the training data to iteratively refine and adjust the centroids of text data. The experimental results reveal that Model-Refinement can dramatically decrease the bias introduced by ECOC, and the combined classifier is comparable to or even better than SVM classifier in performance.

1. Introduction

In recent years, ECOC has been applied to boost the naïve bayes, decision tree and SVM classifier for text data (Berger 1999, Ghani 2000, Ghani 2002, Rennie et al. 2001). Following this research direction, in this work, we explore the use of ECOC to enhance the performance of centroid classifier (Han et al. 2000). To the best of our knowledge, no previous work has been conducted on exactly this problem. The framework we adopted is to decompose one multi-class problem into multiple binary problems and then use centroid classifier to learn the individual binary classification problems.

However, this kind of decomposition incurs considerable bias (Liu et al. 2002) for centroid classifier. In substance, centroid classifier (Han et al. 2000) relies on a simple decision rule that a given document should be assigned a particular class if the similarity (or distance) of this document to the centroid of the class is the largest (or smallest). This decision rule is based on a straightforward assumption that the documents in one category should share some similarities with each other. However, this hypothesis is often violated by ECOC on the grounds that it ignores the similarities of original classes when disassembling one multi-class problem into multiple binary problems.

In order to attack this problem, we use Model-Refinement (Tan et al. 2005) to reduce this so-called bias. The basic idea is to take advantage of misclassified examples in the training data to iteratively refine and adjust the centroids. This technique is very flexible, which only needs one classification method and there is no change to the method in any way.

To examine the performance of proposed method, we conduct an extensive experiment on two commonly used datasets, i.e., Newsgroup and Industry Sector. The results indicate that Model-Refinement can dramatically decrease the bias introduce by ECOC, and the resulted classifier is comparable to or even better than SVM classifier in performance.

2. Error-Correcting Output Coding

Error-Correcting Output Coding (ECOC) is a form of combination of multiple classifiers (Ghani 2000). It works by converting a multi-class supervised learning problem into a large number (L) of two-class supervised learning problems (Ghani 2000). Any learning algorithm that can handle two-class learning problems, such as Naïve Bayes (Sebastiani 2002), can then be applied to learn each of these L problems. L can then be thought of as the length of the codewords

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 81–84,
Prague, June 2007. ©2007 Association for Computational Linguistics

with one bit in each codeword for each classifier. The ECOC algorithm is outlined in Figure 1.

```
TRAINING
1 Load training data and parameters, i.e., the length of code
  L and training class K.
2 Create a L-bit code for the K classes using a kind of
  coding algorithm.
3 For each bit, train the base classifier using the binary
  class (0 and 1) over the total training data.
TESTING
1 Apply each of the L classifiers to the test example.
2 Assign the test example the class with the largest votes.
```

Figure 1: Outline of ECOC

3. Methodology

3.1 The bias incurred by ECOC for centroid classifier

Centroid classifier is a linear, simple and yet efficient method for text categorization. The basic idea of centroid classifier is to construct a centroid C_i for each class c_i using formula (1) where d denotes one document vector and $|z|$ indicates the cardinality of set z. In substance, centroid classifier makes a simple decision rule (formula (2)) that a given document should be assigned a particular class if the similarity (or distance) of this document to the centroid of the class is the largest (or smallest). This rule is based on a straightforward assumption: the documents in one category should share some similarities with each other.

$$C_i = \frac{1}{|c_i|}\sum_{d\in c_i} d \qquad (1)$$

$$c = \arg\max_{c_i}\left(\frac{d\cdot C_i}{\|d\|_2\|C_i\|_2}\right) \qquad (2)$$

For example, the single-topic documents involved with “sport” or “education” can meet with the presumption; while the hybrid documents involved with “sport” as well as “education” break this supposition.

As such, ECOC based centroid classifier also breaks this hypothesis. This is because ECOC ignores the similarities of original classes when producing binary problems. In this scenario, many different classes are often merged into one category. For example, the class “sport” and “education” may be assembled into one class. As a result, the assumption will inevitably be broken.

Let’s take a simple multi-class classification task with 12 classes. After coding the original classes, we obtain the dataset as Figure 2. Class 0 consists of 6 original categories, and class 1 contains another 6 categories. Then we calculate the centroids of merged class 0 and merged class 1 using formula (1), and draw a Middle Line that is the perpendicular bisector of the line between the two centroids.

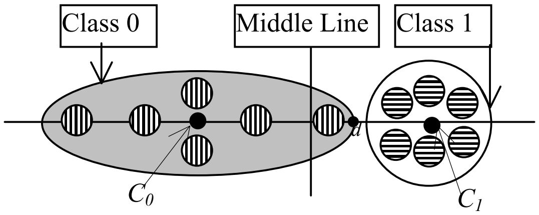


Figure 2: Original Centroids of Merged Class 0 and Class 1

According to the decision rule (formula (2)) of centroid classifier, the examples of class 0 on the right of the Middle Line will be misclassified into class 1. This is the mechanism why ECOC can bring bias for centroid classifier. In other words, the ECOC method conflicts with the assumption of centroid classifier to some degree.

3.2 Why Model-Refinement can reduce this bias?

In order to decrease this kind of bias, we employ the Model-Refinement to adjust the class representative, i.e., the centroids. The basic idea of Model-Refinement is to make use of training errors to adjust class centroids so that the biases can be reduced gradually, and then the training-set error rate can also be reduced gradually.

```
1 Load training data and parameters;
2 Calculate centroid for each class;
3 For iter=1 to MaxIteration Do
  3.1 For each document $d$ in training set Do
        3.1.1 Classify $d$ labeled “$A_1$” into class “$A_2$”;
        3.1.2 If ($A_1$!=$A_2$) Do
          Drag centroid of class $A_1$ to $d$ using formula (3);
          Push centroid of class $A_2$ against $d$ using
          formula (4);
```

Figure 3: Outline of Model-Refinement Strategy

For example, if document d of class 1 is misclassified into class 2, both centroids C_1 and C_2 should be moved right by the following formulas (3-4) respectively,

$$C_1^* = C_1 + \eta\cdot d \qquad (3)$$

$$C_2^* = C_2 - \eta\cdot d \qquad (4)$$

where η ($0<\eta<1$) is the *Learning Rate* which controls the step-size of updating operation.

The Model-Refinement for centroid classifier is outlined in Figure 3 where *MaxIteration* denotes the pre-defined steps for iteration. More details can be found in (Tan et al. 2005). The time requirement of Model-Refinement is $O(MTKW)$ where M denotes the iteration steps.

With this so-called move operation, C_0 and C_1 are both moving right gradually. At the end of this kind of move operation (see Figure 4), no example of class 0 locates at the right of Middle Line so no example will be misclassified.

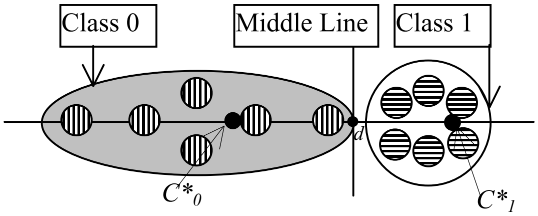


Figure 4: Refined Centroids of Merged Class 0 and Class 1

3.3 The combination of ECOC and Model-Refinement for centroid classifier

In this subsection, we present the outline (Figure 5) of combining ECOC and Model-Refinement for centroid classifier. In substance, the improved ECOC combines the strengths of ECOC and Model-Refinement. ECOC research in ensemble learning techniques has shown that it is well suited for classification tasks with a large number of categories. On the other hand, Model-Refinement has proved to be an effective approach to reduce the bias of base classifier, that is to say, it can dramatically boost the performance of the base classifier.

```
TRAINING
1 Load training data and parameters, i.e., the length of
  code L and training class K.
2 Create a L-bit code for the K classes using a kind of
  coding algorithm.
3 For each bit, train centroid classifier using the binary
  class (0 and 1) over the total training data.
4 Use Model-Refinement approach to adjust centroids.
TESTING
1 Apply each of the L classifiers to the test example.
2 Assign the test example the class with the largest votes.
```

Figure 5: Outline of combining ECOC and Model-Refinement

4. Experiment Results

4.1 Datasets

In our experiment, we use two corpora: NewsGroup[1], and Industry Sector[2].

NewsGroup The NewsGroup dataset contains approximately 20,000 articles evenly divided among 20 Usenet newsgroups. We use a subset consisting of total categories and 19,446 documents.

Industry Sector The set consists of company homepages that are categorized in a hierarchy of industry sectors, but we disregard the hierarchy. There were 9,637 documents in the dataset, which were divided into 105 classes. We use a subset called as Sector-48 consisting of 48 categories and in all 4,581 documents.

4.2 Experimental Design

To evaluate a text classification system, we use MicroF1 and MacroF1 measures (Chai et al. 2002). We employ Information Gain as feature selection method because it consistently performs well in most cases (Yang et al. 1997). We employ TFIDF (Sebastiani 2002) to compute feature weight. For SVM classifier we employ SVMTorch. (www.idiap.ch/~bengio/projects/SVMTorch.html).

4.3 Comparison and Analysis

Table 1 and table 2 show the performance comparison of different method on two datasets when using 10,000 features. For ECOC, we use 63-bit BCH coding; for Model-Refinement, we fix its *MaxIteration* as 8. For brevity, we use MR to denote Model-Refinement.

From the two tables, we can observe that ECOC indeed brings significant bias for centroid classifier, which results in considerable decrease in accuracy. Especially on sector-48, the bias reduces the MicroF1 of centroid classifier from 0.7985 to 0.6422.

On the other hand, the combination of ECOC and Model-Refinement makes a significant performance improvement over centroid classifier.

[1] www-2.cs.cmu.edu/afs/cs/project/theo-11/www/wwkb.

[2] www-2.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/.

On Newsgroup, it beats centroid classifier by 4 percents; on Sector-48, it beats centroid classifier by 11 percents. More encouraging, it yields better performance than SVM classifier on Sector-48. This improvement also indicates that Model-Refinement can effectively reduce the bias incurred by ECOC.

Table 1: The MicroF1 of different methods

Method Dataset	Centroid	MR +Centroid	ECOC +Centroid	ECOC + MR +Centroid	SVM
Sector-48	0.7985	0.8671	0.6422	**0.9122**	0.8948
NewsGroup	0.8371	0.8697	0.8085	**0.8788**	0.8777

Table 2: The MacroF1 of different methods

Method Dataset	Centroid	MR +Centroid	ECOC +Centroid	ECOC + MR +Centroid	SVM
Sector-48	0.8097	0.8701	0.6559	**0.9138**	0.8970
NewsGroup	0.8331	0.8661	0.7936	0.8757	0.8759

Table 3 and 4 report the classification accuracy of combining ECOC with Model-Refinement on two datasets vs. the length BCH coding. For Model-Refinement, we fix its *MaxIteration* as 8; the number of features is fixed as 10,000.

Table 3: the MicroF1 vs. the length of BCH coding

Bit Dataset	15bit	31bit	63bit
Sector-48	0.8461	0.8948	0.9105
NewsGroup	0.8463	0.8745	0.8788

Table 4: the MacroF1 vs. the length of BCH coding

Bit Dataset	15bit	31bit	63bit
Sector-48	0.8459	0.8961	0.9122
NewsGroup	0.8430	0.8714	0.8757

We can clearly observe that increasing the length of the codes increases the classification accuracy. However, the increase in accuracy is not directly proportional to the increase in the length of the code. As the codes get larger, the accuracies start leveling off as we can observe from the two tables.

5. Conclusion Remarks

In this work, we examine the use of ECOC for improving centroid text classifier. The implementation framework is to decompose multi-class problems into multiple binary problems and then learn the individual binary classification problems by centroid classifier. Meanwhile, Model-Refinement is employed to reduce the bias incurred by ECOC.

In order to investigate the effectiveness and robustness of proposed method, we conduct an extensive experiment on two commonly used corpora, i.e., Industry Sector and Newsgroup. The experimental results indicate that the combination of ECOC with Model-Refinement makes a considerable performance improvement over traditional centroid classifier, and even performs comparably with SVM classifier.

References

Berger, A. *Error-correcting output coding for text classification*. In Proceedings of IJCAI, 1999.

Chai, K., Chieu, H. and Ng, H. *Bayesian online classifiers for text classification and filtering*. SIGIR. 2002, 97-104

Ghani, R. *Using error-correcting codes for text classification*. ICML. 2000

Ghani, R. *Combining labeled and unlabeled data for multiclass text categorization*. ICML. 2002

Han, E. and Karypis, G. *Centroid-Based Document Classification Analysis & Experimental Result*. PKDD. 2000.

Liu, Y., Yang, Y. and Carbonell, J. *Boosting to Correct Inductive Bias in Text Classification*. CIKM. 2002, 348-355

Rennie, J. and Rifkin, R. *Improving multiclass text classification with the support vector machine*. In MIT. AI Memo AIM-2001-026, 2001.

Sebastiani, F. *Machine learning in automated text categorization*. ACM Computing Surveys, 2002,34(1): 1-47.

Tan, S., Cheng, X., Ghanem, M., Wang, B. and Xu, H. *A novel refinement approach for text categorization*. CIKM. 2005, 469-476

Yang, Y. and Pedersen, J. *A Comparative Study on Feature Selection in Text Categorization*. ICML. 1997, 412-420.

Poliqarp
An open source corpus indexer and search engine with syntactic extensions

Daniel Janus
Sentivision Polska Sp. z o.o.
Marynarska 19a, 02-674 Warsaw, Poland
nathell@korpus.pl

Adam Przepiórkowski
Insitute of Computer Science
Polish Academy of Sciences
Ordona 21, 01-237 Warsaw, Poland
adamp@ipipan.waw.pl

Abstract

This paper presents recent extensions to Poliqarp, an open source tool for indexing and searching morphosyntactically annotated corpora, which turn it into a tool for indexing and searching certain kinds of treebanks, complementary to existing treebank search engines. In particular, the paper discusses the motivation for such a new tool, the extended query syntax of Poliqarp and implementation and efficiency issues.

1 Introduction

The aim of this paper is to present extensions to Poliqarp,[1] an efficient open source indexer and search tool for morphosyntactically annotated XCES-encoded (Ide et al., 2000) corpora, with query syntax based on that of CQP (Christ, 1994), but extending it in interesting ways. Poliqarp has been in constant development since 2003 (Przepiórkowski et al., 2004) and it is currently employed as the search engine of the IPI PAN Corpus of Polish (Przepiórkowski, 2004) and the Lisbon corpus of Portuguese (Barreto et al., 2006), as well as in other projects. Poliqarp has a typical server-client architecture, with various Poliqarp clients developed so far, including GUI clients for a variaty of operating systems (Linux, Windows, MacOS, Solaris) and architectures (big-endian and little-endian), as well as a PHP client. Since March 2006, the 1st stable version of Poliqarp (Janus and Przepiórkowski, 2006) is available under GPL.[2] A version of Poliqarp that implements various statistical extensions is at the beta-testing stage.

Although Poliqarp was designed as a tool for corpora linguistically annotated at word-level only, the extensions described in this paper turn it into an indexing and search tool for certain kinds of treebanks, complementary to existing treebank search engines.

Section 2 briefly introduces the basic query syntax of Poliqarp, section 3 presents extensions of Poliqarp aimed at the processing of treebanks, section 4 discusses implementation and efficiency issues, and section 5 concludes the paper.

2 Query Syntax

In the Poliqarp query language, just as in CQP, regular expressions may be formulated over corpus positions, e.g.: `[pos="adj"]+`, where any non-empty sequence of adjectives is sought, or within values of attributes, e.g.: `[pos="a.*"]`, concerning forms (henceforth: segments) tagged with POSs whose names start with an `a`, e.g., `adj` and `adv`.

Parts of speech and morphosyntactic categories may be queried separately, e.g., the query `[gend=masc]` could be used to search for masculine segments, regardless of the POS or other categories, while the query `[pos="subst|ger"&gend!=masc]` can be used to find nominal and gerundive segments which are not masculine.

A unique feature of Poliqarp is that it may be used for searching corpora containing, in addition to disambiguated interpretations, information about all

[1] **Pol**yinterpretation **I**ndexing **Q**uery **A**nd **R**etrieval **P**rocessor

[2] Cf. `http://poliqarp.sourceforge.net/`.

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 85–88,
Prague, June 2007. ©2007 Association for Computational Linguistics

possible morphosyntactic interpretations given by the morphological analyser. For example, the query `[case~acc]` finds all segments with an accusative interpretation (even if this is not the interpretation selected in a given context), while `[case=acc]` finds segments which were disambiguated to accusative in a given context.

Moreover, Poliqarp does not make the assumption that only one interpretation must be correct for any given segment; some examples of sentences containing an ambiguous segment which cannot be uniquely disambiguated even given unlimited context and all the linguistic and encyclopaedic knowledge are cited in (Przepiórkowski et al., 2004). In such cases, the = operator has the existential meaning, i.e., `[case=acc]` finds segments with at least one accusative interpretation marked as correct in the context (“disambiguated”). On the other hand, the operator == is universal, i.e., `[case==acc]` finds segments whose all disambiguated interpretations are accusative: segments which were truly uniquely disambiguated to one (accusative) interpretation, or segments which have many interpretations correct in the context, but all of them are accusative.[3] For completeness, the operator ~~ is added, which universally applies to all morphosyntactic interpretations, i.e., `[case~~acc]` finds segments whose all interpretations as given by a morphological analyser (before disambiguation) are accusative.

The most detailed presentation of the original query syntax of Poliqarp is available in (Przepiórkowski, 2004), downloadable from `http://korpus.pl/index.php?page=publications`.

3 Syntactic Extensions

(Przepiórkowski, 2007) argues for the explicit representation of both a syntactic head and a semantic head for each syntactic group identified in a (partially parsed) constituency-based (as opposed to dependency-based) treebank. For example, for the Polish syntactic group *tuzin białych koni*, ‘a dozen of white horses’, lit. ‘dozen-NOM/ACC white-GEN horses-GEN’, the syntactic head is *tuzin* ‘dozen’, while the semantic head is *koni* ‘horses’. The segment *koni* is also both the syntactic head and the semantic head of the embedded nominal group *białych koni* ‘white horses’. In general, following (Przepiórkowski, 2007), a given segment is a syntactic head of at most one group (e.g., *tuzin* and *koni* in the example above), but it may be a semantic head of a number of groups (e.g., *koni* above is a semantic head of *białych koni* and of *tuzin białych koni*).

This kind of representation is problematic for general search tools for constituency-based treebanks,[4] such as TIGERSearch (Lezius, 2002),[5] which usually assume that the set of edges within a syntactic representation of a sentence is a tree, in particular, that it has a single root node and that each leaf has (at most) one incoming edge.[6] While the former assumption is not a serious problem (an artificial single root may always be added), the latter is fatal for representations alluded to above, as a single segment may be a semantic head of a number of syntactic groups, i.e., it may have several incoming edges.

The extension of Poliqarp presented here makes it possible to index and search for such (partial) syntactic-semantic treebanks. Specifications of syntactic constructions in the extended Poliqarp query language syntax are similar to specifications of particular segments, but they use a different repertoire of attributes, non-overlapping with the attributes used to specify single segments. Two main attributes to be used for querying for syntactic groups are: `type` and `head`. The attribute `type` specifies the general syntactic type of the group, so `[type=Coordination]` will find coordinated constructions, while `[type="[PN]G"]` will find prepositional and nominal groups.

The syntax of values of the attribute `head` differs from that of the other attributes; its values must be enclosed in a double or a single set of square brackets, as in: `[head=[...][...]]` or `[head=[...]]`. In the first case, the first brackets specify the syntactic head and second brackets specify the semantic

[3] In Polish this may happen, for example, in case of some gerund forms which are homographs of true nouns, where meaning does not make it possible to decide on the nominal / gerundive interpretation of the form.

[4] It seems that it would also be problematic for dependency tools such as Netgraph, cf. (Hajič et al., 2006) and `http://quest.ms.mff.cuni.cz/netgraph/doc/netgraph_manual.html`.

[5] Cf. `http://www.ims.uni-stuttgart.de/projekte/TIGER/`.

[6] In TIGER tools, there is a special mechanism for adding a second edge, e.g., in order to represent control.

head, as in the following query which may be used to find elective constructions of the type *najstarszy z koni* '(the) oldest of horses', which are syntactically headed by the adjective and semantically by the semantic head of the dependent of that adjective: `[head=[pos=adj][pos=noun]]`.

In the second case, the content of the single brackets specifies both the syntactic head and the semantic head and, additionally, makes the requirement that they be the same segment. This means that the queries `[head=[case=gen][case=gen]]` and `[head=[case=gen]]` have a slightly different semantics: the first will find syntactic groups where the two heads may be different or the same, but they must be genitive; the second will find groups with the two heads being necessarily the same genitive segment.

The usefulness of such queries may be illustrated with a query for verbs which co-occur with dative dependents denoting students; the first approximation of such a query may look like this: `[pos=verb][head=[case=dat][base=student]]`. This query will find not only dative nominal groups headed by a form of STUDENT, but also dative numeral groups whose main noun is a form of STUDENT, appropriate dative adjectival elective groups, etc.

As syntactic sugar, the constructs `synh=[...]` and `semh=[...]` can be used to enforce a constraint only on, respectively, syntactic or semantic head of a group.

It may seem that, given the possibility to specify the syntactic head of the construction, the attribute `type` is redundant; in fact, we are not currently aware of cases where the specification `type="PG"` or `type="NG"` could not be replaced by an appropriate reference to the grammatical class (part of speech) of the syntactic head. However, the `type` attribute is useful for finding constructions which are not defined by their heads, for example, *oratio recta* constructions, and it is also useful for dealing with coordinate structures.

4 Implementation Issues

To allow for fast searching, the original Poliqarp uses its own compact binary format for corpora, described in detail in (Janus, 2006) and briefly in (Janus and Przepiórkowski, 2006). Because the number of syntactic groups can easily grow very large and be on par with total number of words in a fully-tagged corpus, the representation of syntactic groups should be space-efficient, yet allow for fast decoding and random access.

The key observation to achieving this goal is that, due to the tree nature of the group set, any two groups can be either mutually disjoint or completely contained in each other. Thus, it is possible to serialize the tree into a list, sorted by the lower bound of a group,[7] such that each group is immediately followed by its direct subgroups.

More precisely, the on-disk representation of a treebank is a bit vector that contains the following data for each group: 1) synchronization bit (see below), usually 0; 2) the difference between the lower bound of the previous group and the lower bound of the one in question, encoded in γ-code;[8] 3) γ-encoded length of current group in segments; 4) γ-encoded number of type of this group (the mapping of numbers to type names is stored in a separate on-disk dictionary in which two type numbers are reserved: 0 for coordinated groups and 1 for conjunctions); 5) if this is a coordinated construct (i.e., $type = 0$) — γ-encoded number of subsequent groups (excluding the current one but including indirect subgroups) that are part of the coordination;[9] or 6) if this is not a coordinated construct (i.e., it is an ordinary group) — offset of syntactic and semantic head of this group, in that order, each represented by a binary number of $\log l$ bits, where l stands for the length of the group.

One drawback of this representation is that it does not allow for random access: the γ-code and head offsets have variable length, thus it is not possible to determine which bit one should start with to decode the group sequence for a certain segment. To mitigate this, a synchronization mechanism is employed.

[7] The corpus proper is represented by one large vector of fixed-size structures denoting segments; here, the bounds of a group mean offsets into that vector.

[8] The γ-code is a prefix-free variable-length code that encodes arbitrary integers so that the representation of small numbers takes few bits; see (Witten et al., 1999) for details.

[9] Special treatment of coordination is caused by the fact that, as argued in (Przepiórkowski, 2007), coordinate structures are best treated as multi-headed constructions, with each conjunct bringing its own syntactic and semantic head.

For every k-th segment (k is a constant defined for the corpus, usually 1024), the bit offset of start of the description of the earliest group that intersects this segment is stored as an unsigned little-endian 32-bit integer in a separate file. In the description of this group, the synchronization bit is set to 1, and the lower bound is spelled in full (as an unsigned 32-bit binary integer) so that it is not necessary to know the previous lower bound to start decoding.

This synchronization lines up with the sparse inverted indexing mechanism used by Poliqarp for efficient searching. Poliqarp artificially splits the corpus into fixed-size chunks and remembers which segments occur in which chunks; if the search engine makes random access to the corpus, the accessed segments' offsets are multiplies of the chunk size. It is best, thus, to ascertain that the constant k is also equal to this chunk size.

In a typical scenario with many mostly small groups occurring close to each other, this encoding schema is capable of achieving the ratio of well under two bytes per group and does not incur a significant overhead in corpus size (which is usually in the range of 10–12 bytes times the number of segments for a morphosyntactically but not structurally tagged corpus). This is important, since disk access is the key factor in Poliqarp's performance.

5 Conclusions

In this paper, we presented an extension of Poliqarp, a tool for indexing and searching morphosyntactically annotated corpora, towards the management of syntactically annotated corpora. An interesting feature of thus extended Poliqarp is its ability to deal with treebanks which do not adopt the "at most one incoming edge" assumption and which distinguish between syntactic heads and semantic heads. We also sketched the original and efficient method of indexing such treebanks. The implementation of the extensions currently approaches the alpha stage. By the time of ACL 2007, we expect to release the sources of a relatively stable beta-stage version.

References

Florbela Barreto, António Branco, Eduardo Ferreira, Amália Mendes, Maria Fernanda Nascimento, Filipe Nunes, and João Silva. 2006. Open resources and tools for the shallow processing of Portuguese: The TagShare project. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC 2006)*.

Oli Christ. 1994. A modular and flexible architecture for an integrated corpus query system. In *COMPLEX'94*, Budapest.

Jan Hajič, Eva Hajičová, Jaroslava Hlaváčová, Václav Klimeš, Jiří Mirovský, Petr Pajas, Jan Štěpánek, Barbara Vidová Hladká, and Zdeněk Žabokrtský, 2006. *PDT 2.0 – Guide*. Charles University, Prague. June 20, 2006.

Nancy Ide, Patrice Bonhomme, and Laurent Romary. 2000. XCES: An XML-based standard for linguistic corpora. In *Proceedings of the Linguistic Resources and Evaluation Conference*, pages 825–830, Athens, Greece.

Daniel Janus and Adam Przepiórkowski. 2006. Poliqarp 1.0: Some technical aspects of a linguistic search engine for large corpora. In Jacek Waliński, Krzysztof Kredens, and Stanisław Goźdź-Roszkowski, editors, *The proceedings of Practical Applications of Linguistic Corpora 2005*, Frankfurt am Main. Peter Lang.

Daniel Janus. 2006. Metody przeszukiwania i obrazowania jego wyników w dużych korpusach tekstów. Master's thesis, Uniwersytet Warszawski, Wydział Matematyki, Informatyki i Mechaniki, Warsaw.

Wolfgang Lezius. 2002. TIGERSearch — ein Suchwerkzeug für Baumbanken. In Stephan Busemann, editor, *Proceedings der 6. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS 2002)*, Saarbrücken.

Adam Przepiórkowski, Zygmunt Krynicki, Łukasz Dębowski, Marcin Woliński, Daniel Janus, and Piotr Bański. 2004. A search tool for corpora with positional tagsets and ambiguities. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation, LREC 2004*, pages 1235–1238, Lisbon. ELRA.

Adam Przepiórkowski. 2004. *The IPI PAN Corpus: Preliminary version*. Institute of Computer Science, Polish Academy of Sciences, Warsaw.

Adam Przepiórkowski. 2007. On heads and coordination in valence acquisition. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing (CICLing 2007)*, Lecture Notes in Computer Science, pages 50–61, Berlin. Springer-Verlag.

Ian H. Witten, Alistair Moffat, and Timothy C. Bell. 1999. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishers Inc., San Francisco, CA, 2nd edition.

Test Collection Selection and Gold Standard Generation for a Multiply-Annotated Opinion Corpus

Lun-Wei Ku, Yong-Shen Lo and Hsin-Hsi Chen
Department of Computer Science and Information Engineering
National Taiwan University
{lwku, yslo}@nlg.csie.ntu.edu.tw; hhchen@csie.ntu.edu.tw

Abstract

Opinion analysis is an important research topic in recent years. However, there are no common methods to create evaluation corpora. This paper introduces a method for developing opinion corpora involving multiple annotators. The characteristics of the created corpus are discussed, and the methodologies to select more consistent testing collections and their corresponding gold standards are proposed. Under the gold standards, an opinion extraction system is evaluated. The experiment results show some interesting phenomena.

1 Introduction

Opinion information processing has been studied for several years. Researchers extracted opinions from words, sentences, and documents, and both rule-based and statistical models are investigated (Wiebe *et al*., 2002; Pang *et al*., 2002). The evaluation metrics precision, recall and f-measure are usually adopted.

A reliable corpus is very important for the opinion information processing because the annotations of opinions concern human perspectives. Though the corpora created by researchers were analyzed (Wiebe *et al*., 2002), the methods to increase the reliability of them were seldom touched. The strict and lenient metrics for opinions were mentioned, but not discussed in details together with the corpora and their annotations.

This paper discusses the selection of testing collections and the generation of the corresponding gold standards under multiple annotations. These testing collections are further used in an opinion extraction system and the system is evaluated with the corresponding gold standards. The analysis of human annotations makes the improvements of opinion analysis systems feasible.

2 Corpus Annotation

Opinion corpora are constructed for the research of opinion tasks, such as opinion extraction, opinion polarity judgment, opinion holder extraction, opinion summarization, opinion question answering, etc.. The materials of our opinion corpus are news documents from NTCIR CIRB020 and CIRB040 test collections. A total of 32 topics concerning opinions are selected, and each document is annotated by three annotators. Because different people often feel differently about an opinion due to their own perspectives, multiple annotators are necessary to build a reliable corpus. For each sentence, whether it is relevant to a given topic, whether it is an opinion, and if it is, its polarity, are assigned. The holders of opinions are also annotated. The details of this corpus are shown in Table 1.

	Topics	Documents	Sentences
Quantity	32	843	11,907

Table 1. Corpus size

3 Analysis of Annotated Corpus

As mentioned, each sentence in our opinion corpus is annotated by three annotators. Although this is a must for building reliable annotations, the inconsistency is unavoidable. In this section, all the possible combinations of annotations are listed and two methods are introduced to evaluate the quality of the human-tagged opinion corpora.

3.1 Combinations of annotations

Three major properties are annotated for sentences in this corpus, i.e., the relevancy, the opinionated issue, and the holder of the opinion. The combinations of relevancy annotations are simple, and annotators usually have no argument over the opinion holders. However, for the annotation of the opinionated issue, the situation is more com-

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 89–92,
Prague, June 2007. ©2007 Association for Computational Linguistics

plex. Annotations may have an argument about whether a sentence contains opinions, and their annotations may not be consistent on the polarities of an opinion. Here we focus on the annotations of the opinionated issue. Sentences may be considered as opinions only when more than two annotators mark them opinionated. Therefore, they are targets for analysis. The possible combinations of opinionated sentences and their polarity are shown in Figure 1.

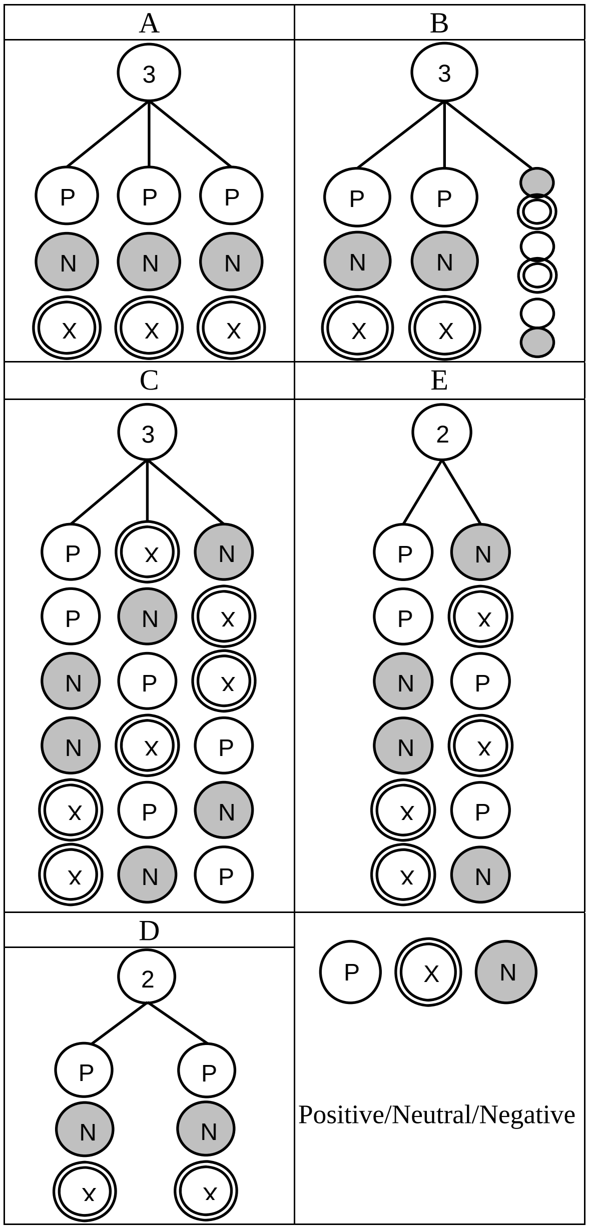


Figure 1. Possible combinations of annotations

In Figure 1, Cases A, B, C are those sentences which are annotated as opinionated by all three annotators, while cases D, E are those sentences which are annotated as opinionated only by two annotators. In case A and case D, the polarities annotated by annotators are identical. In case B, the polarities annotated by two of three annotators are agreed. However, in cases C and E, the polarities annotated disagree with each other. The statistics of these five cases are shown in Table 2.

Case	A	B	C	D	E	All
Number	1,660	1,076	124	2,413	1,826	7,099

Table 2. Statistics of cases A-E

3.2 Inconsistency

Multiple annotators bring the inconsistency. There are several kinds of inconsistency in annotations, for example, relevant/non-relevant, opinionated/non-opinionated, and the inconsistency of polarities. The relevant/non-relevant inconsistency is more like an information retrieval issue. For opinions, because their strength varies, sometimes it is hard for annotators to tell if a sentence is opinionated. However, for the opinion polarities, the inconsistency between positive and negative annotations is obviously stronger than that between positive and neutral, or neutral and negative ones. Here we define a sentence “strongly inconsistent” if both positive and negative polarities are assigned to a sentence by different annotators. The strong inconsistency may occur in case B (171), C (124), and E (270). In the corpus, only about 8% sentences are strongly inconsistent, which shows the annotations are reliable.

3.3 Kappa value for agreement

We further assess the usability of the annotated corpus by Kappa values. Kappa value gives a quantitative measure of the magnitude of inter-annotator agreement. Table 3 shows a commonly used scale of the Kappa values.

Kappa value	Meaning
<0	less than change agreement
0.01-0.20	slight agreement
0.21-0.40	fair agreement
0.41-0.60	moderate agreement
0.61-0.80	substantial agreement
0.81-0.99	almost perfect agreement

Table 3. Interpretation of Kappa value

The inconsistency of annotations brings difficulties in generating the gold standard. Sentences should first be selected as the testing collection, and then the corresponding gold standard can be generated. Our aim is to generate testing collections and their gold standards which agree mostly to annotators. Therefore, we analyze the kappa value not between annotators, but between the annotator and the gold standard. The methodologies are introduced in the next section.

4 Testing Collections and Gold Standards

The gold standard of relevance, the opinionated issue, and the opinion holder must be generated according to all the annotations. Answers are chosen based on the agreement of annotations. Considering the agreement among annotations themselves, the strict and the lenient testing collections and their corresponding gold standard are generated. Considering the Kappa values of each annotator and the gold standard, topics with high agreement are selected as the testing collection. Moreover, considering the consistency of polarities, the substantial consistent testing collection is generated. In summary, two metrics for generating gold standards and four testing collections are adopted.

4.1 Strict and lenient

Namely, the strict metric is different from the lenient metric in the agreement of annotations. For the strict metric, sentences with annotations agreed by all three annotators are selected as the testing collection and the annotations are treated as the strict gold standard; for the lenient metric, sentences with annotations agreed by at least two annotators are selected as the testing collection and the majority of annotations are treated as the lenient gold standard. For example, for the experiments of extracting opinion sentences, sentences in cases A, B, and C in Figure 1 are selected in both strict and lenient testing collections, while sentences in cases D and E are selected only in the lenient testing collection because three annotations are not totally agreed with one another. For the experiments of opinion polarity judgment, sentences in case A in Figure 1 are selected in both strict and lenient testing collections, while sentences in cases B, C, D and E are selected only in the lenient testing collection. Because every opinion sentence should be given a polarity, the polarities of sentences in cases B and D are the majority of annotations, while the polarity of sentences in cases C are given the polarity neutral in the lenient gold standard. The polarities of sentences in case E are decided by rules P+X=P, N+X=N, and P+N=X. As for opinion holders, holders are found in opinion sentences of each testing collection. The strict and lenient metrics are also applied in annotations of relevance.

4.2 High agreement

To see how the generated gold standards agree with the annotations of all annotators, we analyze the kappa value from the agreements of each annotator and the gold standard for all 32 topics. Each topic has two groups of documents from NTCIR: very relevant and relevant to topic. However, one topic has only the relevant type document, it results in a total of 63 (2*31+1) groups of documents. Note that the lenient metric is applied for generating the gold standard of this testing collection because the strict metric needs perfect agreement with each annotator's annotations. The distribution of kappa values of 63 groups is shown in Table 4 and Table 5. The cumulative frequency bar graphs of Table 4 and Table 5 are shown in Figure 2 and Figure 3.

Kappa	<=0	0-0.2	0.21-0.4	0.41-0.6	0.61-0.8	0.81-0.99
Number	1	2	12	14	33	1

Table 4. Kappa values for opinion extraction

Kappa	<=0	0-0.2	0.21-0.4	0.41-0.6	0.61-0.8	0.81-0.99
Number	9	0	7	21	17	9

Table 5. Kappa values for polarity judgment

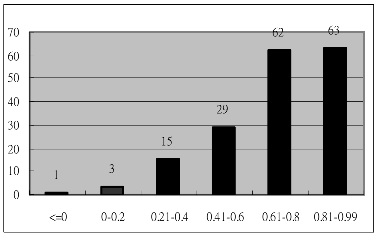


Figure 2. Cumulative frequency of Table 4

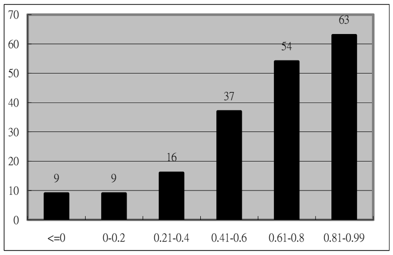


Figure 3. Cumulative frequency of Table 5

According to Figure 2 and Figure 3, document groups with kappa values above 0.4 are selected as the high agreement testing collection, that is, document groups with moderate agreement in Table 3. A total of 48 document groups are collected for opinion extraction and 47 document groups are collected for opinion polarity judgment.

4.3 Substantial Consistency

In Section 3.2, sentences which are “strongly inconsistent” are defined. The substantial consistency test collection expels strongly inconsistent sentences to achieve a higher consistency. Notice that this test collection is still less consistent than the strict test collection, which is perfectly consistent with annotators. The lenient metric is applied for generating the gold standard for this collection.

5 An Opinion System -- CopeOpi

A **C**hinese **op**inion **e**xtraction system for **op**inionated **i**nformation, CopeOpi, is introduced here. (Ku *et al.*, 2007) When judging the opinion polarity of a sentence in this system, three factors are considered: sentiment words, negation operators and opinion holders. Every sentiment word has its own sentiment score. If a sentence consists of more positive sentiments than negative sentiments, it must reveal something good, and vice versa. However, a negation operator, such as ”not” and ”never”, may totally change the sentiment polarity of a sentiment word. Therefore, when a negation operator appears together with a sentiment word, the opinion score of the sentiment word S will be changed to -S to keep the strength but reverse the polarity. Opinion holders are also considered for opinion sentences, but how they influence opinions has not been investigated yet. As a result, they are weighted equally at first. A word is considered an opinion holder of an opinion sentence if either one of the following two criteria is met:

1. The part of speech is a person name, organization name or personal.
2. The word is in class A (human), type Ae (job) of the Cilin Dictionary (Mei *et al.*, 1982).

6 Evaluation Results and Discussions

Experiment results of CopeOpi using four designed testing collections are shown in Table 6. Under the lenient metric with the lenient test collection, f-measure scores 0.761 and 0.383 are achieved by CopeOpi. The strict metric is the most severe, and the performance drops a lot under it. Moreover, when using high agreement (H-A) and substantial consistency (S-C) test collections, the performance of the system does not increase in portion to the increase of agreement. According to the agreement of annotators, people should perform best in the strict collection, and both high agreement and substantial consistency testing collections are easier than the lenient one. This phenomenon shows that though this system’s performance is satisfactory, its behavior is not like human beings. For a computer system, the lenient testing collection is fuzzier and contains more information for judgment. However, this also shows that the system may only take advantage of the surface information. If we want our systems really judge like human beings, we should enhance the performance on strict, high agreement, and substantial consistency testing collections. This analysis gives us, or other researchers who use this corpus for experiments, a direction to improve their own systems.

	Opinion Extraction			Opinion + Polarity		
Measure	P	R	F	P	R	F
Lenient	0.664	0.890	0.761	0.335	0.448	0.383
Strict	0.258	0.921	0.404	0.104	0.662	0.180
H-A	0.677	0.885	0.767	0.339	0.455	0.388
S-C				0.308	0.452	0.367

Table 6. Evaluation results

Acknowledgments

Research of this paper was partially supported by Excellent Research Projects of National Taiwan University, under the contract 95R0062-AE00-02.

References

Mei, J., Zhu, Y. Gao, Y. and Yin, H.. *tong2yi4ci2ci2lin2*. Shanghai Dictionary Press, 1982.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up? Sentiment classification using machine learning techniques. *Proceedings of the 2002 Conference on EMNLP*, pages 79-86.

Wiebe, J., Breck, E., Buckly, C., Cardie, C., Davis, P., Fraser, B., Litman, D., Pierce, D., Riloff, E., and Wilson, T. (2002). NRRC summer workshop on multi-perspective question answering, final report. *ARDA NRRC Summer 2002 Workshop*.

Ku, L.-W., Wu, T.-H., Li, L.-Y. and Chen., H.-H. (2007). Using Polarity Scores of Words for Sentence-level Opinion Extraction. *Proceedings of the Sixth NTCIR Workshop*.

Generating Usable Formats for Metadata and Annotations in a Large Meeting Corpus

Andrei Popescu-Belis and Paula Estrella
ISSCO/TIM/ETI, University of Geneva
40, bd. du Pont-d'Arve
1211 Geneva 4 - Switzerland
{andrei.popescu-belis, paula.estrella}@issco.unige.ch

Abstract

The AMI Meeting Corpus is now publicly available, including manual annotation files generated in the NXT XML format, but lacking explicit metadata for the 171 meetings of the corpus. To increase the usability of this important resource, a representation format based on relational databases is proposed, which maximizes informativeness, simplicity and reusability of the metadata and annotations. The annotation files are converted to a tabular format using an easily adaptable XSLT-based mechanism, and their consistency is verified in the process. Metadata files are generated directly in the IMDI XML format from implicit information, and converted to tabular format using a similar procedure. The results and tools will be freely available with the AMI Corpus. Sharing the metadata using the Open Archives network will contribute to increase the visibility of the AMI Corpus.

1 Introduction

The AMI Meeting Corpus (Carletta and al., 2006) is one of the largest and most extensively annotated data sets of multimodal recordings of human interaction. The corpus contains 171 meetings, in English, for a total duration of ca. 100 hours. The meetings either follow the remote control design scenario, or are naturally occurring meetings. In both cases, they have between 3 and 5 participants.

Perhaps the most valuable resources in this corpus are the high quality annotations, which can be used to train and test NLP tools. The existing annotation dimensions include, beside transcripts, forced temporal alignment, named entities, topic segmentation, dialogue acts, abstractive and extractive summaries, as well as hand and head movement and posture. However, these dimensions as well as the implicit metadata for the corpus are difficult to exploit by NLP tools due to their particular coding schemes.

This paper describes work on the generation of annotation and metadata databases in order to increase the usability of these components of the AMI Corpus. In the following sections we describe the problem, present the current solutions and give future directions.

2 Description of the Problem

The AMI Meeting Corpus is publicly available at `http://corpus.amiproject.org` and contains the following media files: audio (headset mikes plus lapel, array and mix), video (close up, wide angle), slides capture, whiteboard and paper notes. In addition, all annotations described in Section 1 are available in one large bundle. Annotators followed dimension-specific guidelines and used the NITE XML Toolkit (NXT) to support their task, generating annotations in NXT format (Carletta and al., 2003; Carletta and Kilgour, 2005). Using the NXT/XML schema makes the annotations consistent along the corpus but more difficult to use without the NITE toolkit. A less developed aspect of the corpus is the metadata encoding all auxiliary information about meetings in a more structured and informative manner. At the moment, metadata is spread implicitly along the corpus data, for example

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 93–96,
Prague, June 2007. ©2007 Association for Computational Linguistics

it is encoded in the file or folder names or appears to be split in several resource files.

We define here annotations as the time-dependent information which is abstracted from the input media, i.e. “higher-level” phenomena derived from low-level mono- or multi-modal features. Conversely, metadata is defined as the static information about a meeting that is not directly related to its content (see examples in Section 4). Therefore, though not necessarily time-dependent, structural information derived from meeting-related documents would constitute an annotation and not metadata. These definitions are not universally accepted, but they allow us to separate the two types of information.

The main goal of the present work is to facilitate the use of the AMI Corpus metadata and annotations as part of the larger objective of automating the generation of annotation and metadata databases to enhance search and browsing of meeting recordings. This goal can be achieved by providing plug-and-play databases, which are much easier to access than NXT files and provide declarative rather than implicit metadata. One of the challenges in the NXT-to-database conversion is the extraction of relevant information, which is done here by solving NXT pointers and discarding NXT-specific markup to group all information for a phenomenon in only one structure or table.

The following criteria were important when defining the conversion procedure and database tables:

- Simplicity: the structure of the tables should be easy to understand, and should be close to the annotation dimensions—ideally one table per annotation. Some information can be duplicated in several tables to make them more intelligible. This makes the update of this information more difficult, but as this concerns a recorded corpus, changes are less likely to occur; if such changes do occur, they would first be input in the annotation files, from which a new set of tables can easily be generated.
- Reusability: the tools allow anyone to recreate the tables from the official distribution of the annotation files. Therefore, if the format of the annotation files or folders changes, or if a different format is desired for the tables, it is quite easy to change the tools to generate a new version of the database tables.
- Applicability: the tables are ready to be loaded into any SQL database, so that they can be immediately used by a meeting browser plugged into the database.

Although we report one solution here, there are other approaches to the same problem relying, for example, on different database structures using more or fewer tables to represent this information.

3 Annotations: Generation of Tables

The first goal is to convert the NXT files from the AMI Corpus into a compact tabular representation (tab-separated text files), using a simple, declarative and easily updatable conversion procedure.

The conversion principle is the following: for each type of annotation, which is generally stored in a specific folder of the data distribution, an XSLT stylesheet converts the NXT XML file into a tab-separated text file, possibly using information from one or more annotations. The stylesheets resolve most of the NXT pointers, by including redundant information into the tables, in order to speed up queries by avoiding frequent joins. A Perl script applies the respective XSLT stylesheet to each annotation file according to its type, and generates the global tab-separated files for each annotation. The script also generates an SQL script that creates a relational annotation database and populates it with data from the tab-separated files. The Perl script also summarizes the results into a log file named `<timestamp>.log`.

The conversion process can be summarized as follows and can be repeated at will, in particular if the NXT source files are updated:

1. Start with the official NXT release (or other XML-based format) of the AMI annotations as a reference version.
2. Apply the table generation mechanism to XML annotation files, using XSLT stylesheets called by the script, in order to generate tabular files (TSV) and a table-creation script (`db_loader.sql`).
3. Create and populate the annotation database.
4. Adapt the XSLT stylesheets as needed for various annotations and/or table formats.

4 Metadata: Generation of Explicit Files and Conversion to Tabular Format

As mentioned in Section 2, metadata denotes here any *static information* about a meeting, not directly related to its content. The main metadata items are: date, time, location, scenario, participants, participant-related information (codename, age, gender, knowledge of English and other languages), relations to media-files (participants vs. audio channels vs. files), and relations to other documents produced during the meeting (slides, individual and whiteboard notes).

This important information is spread in many places, and can be found as attributes of a meeting in the annotation files (e.g. start time) or obtained by parsing file names (e.g. audio channel, camera). The relations to media files are gathered from different resource files: mainly the `meetings.xml` and `participants.xml` files. An additional problem in reconstructing such relations (e.g. files generated by a specific participant) is that information about the media resources must be obtained directly from the AMI Corpus distribution web site, since the media resources are not listed explicitly in the annotation files. This implies using different strategies to extract the metadata: for example, stylesheets are the best option to deal with the above-mentioned XML files, while a crawler script is used for HTTP access to the distribution site. However, the solution adopted for annotations in Section 3 can be reused with one major extension and applied to the construction of the metadata database.

The standard chosen for the explicit metadata files is the IMDI format, proposed by the ISLE Meta Data Initiative (Wittenburg et al., 2002; Broeder et al., 2004a) (see `http://www.mpi.nl/IMDI/tools`), which is precisely intended to describe multimedia recordings of dialogues. This standard provides a flexible and extensive schema to store the defined metadata either in specific IMDI elements or as additional key/value pairs. The metadata generated for the AMI Corpus can be explored with the IMDI BC-Browser (Broeder et al., 2004b), a tool that is freely available and has useful features such as search or metadata editing.

The process of extracting, structuring and storing the metadata is as follows:

1. Crawl the AMI Corpus website and store resulting metadata (related to media files) into an XML auxiliary file.
2. Apply an XSLT stylesheet to the auxiliary XML file, using also the distribution files `meetings.xml` and `participants.xml`, to obtain one IMDI file per meeting.
3. Apply the table generation mechanism to each IMDI file in order to generate tabular files (TSV) and a table-creation script.
4. Create and populate metadata tables within database.
5. Adapt the XSLT stylesheet as needed for various table formats.

5 Results: Current State and Distribution

The 16 annotation dimensions from the public AMI Corpus were processed following the procedure described in Section 3. The main Perl script, `anno-xml2db.pl`, applied the 16 stylesheets corresponding to each annotation dimension, which generated one large tab-separated file each. The script also generated the table-creation SQL script `db_loader.sql`. The number of lines of each table, hence the number of "elementary annotations", is shown in Table 1.

The application of the metadata extraction tools described in Section 4 generated a first version of the explicit metadata for the AMI Corpus, consisting of 171 automatically generated IMDI files (one per meeting). In addition, 85 manual files were created in order to organize the metadata files into IMDI corpus nodes, which form the skeleton of the corpus metadata and allow its browsing with the BC-Browser. The resources and tools for annotation/metadata processing will be made soon available on the AMI Corpus website, along with a demo access to the BC-Browser.

6 Discussion and Perspectives

The proposed solution for annotation conversion is easy to understand, as it can be summarized as "one table per annotation dimension". The tables preserve only the relevant information from the NXT

Annotation dimension	Nb. of entries
words (transcript)	1,207,769
named entities	14,230
speech segments	69,258
topics	1,879
dialogue acts	117,043
adjacency pairs	26,825
abstractive summaries	2,578
extractive summaries	19,216
abs/ext links	22,101
participant summaries	3,409
focus	31,271
hand gesture	1,453
head gesture	36,257
argument structures	6,920
argumentation relations	4,759
discussions	8,637

Table 1: Results of annotation conversion; dimensions are grouped by conceptual similarity.

annotation files, and search is accelerated by avoiding repeated joins between tables.

The process of metadata extraction and generation is very flexible and the obtained data can be easily stored in different file formats (e.g. tab-separated, IMDI, XML, etc.) with no need to repeatedly parse file names or analyse folders. Moreover, the advantage of creating IMDI files is that the metadata is compliant with a widely used standard accompanied by freely available tools such as the metadata browser. These results will also help disseminating the AMI Corpus.

As a by-product of the development of annotation and metadata conversion tools, we performed a consistency checking and reported a number of to the corpus administrators. The automatic processing of the entire annotation and metadata set enabled us to test initial hypotheses about annotation structure.

In the future we plan to include the AMI Corpus metadata in public catalogues, through the Open (Language) Archives Initiatives network (Bird and Simons, 2001), as well as through the IMDI network (Wittenburg et al., 2004). The metadata repository will be harvested by answering the OAI-PMH protocol, and the AMI Corpus website could become itself a metadata provider.

Acknowledgments

The work presented here has been supported by the Swiss National Science Foundation through the NCCR IM2 on Interactive Multimodal Information Management (`http://www.im2.ch`). The authors would like to thank Jean Carletta, Jonathan Kilgour and Maël Guillemot for their help in accessing the AMI Corpus.

References

Steven Bird and Gary Simons. 2001. Extending Dublin Core metadata to support the description and discovery of language resources. *Computers and the Humanities*, 37(4):375–388.

Daan Broeder, Thierry Declerck, Laurent Romary, Markus Uneson, Sven Strömqvist, and Peter Wittenburg. 2004a. A large metadata domain of language resources. In *LREC 2004 (4th Int. Conf. on Language Resources and Evaluation)*, pages 369–372, Lisbon.

Daan Broeder, Peter Wittenburg, and Onno Crasborn. 2004b. Using profiles for IMDI metadata creation. In *LREC 2004 (4th Int. Conf. on Language Resources and Evaluation)*, pages 1317–1320, Lisbon.

Jean Carletta and al. 2006. The AMI Meeting Corpus: A pre-announcement. In Steve Renals and Samy Bengio, editors, *Machine Learning for Multimodal Interaction II*, LNCS 3869, pages 28–39. Springer-Verlag, Berlin/Heidelberg.

Jean Carletta and Jonathan Kilgour. 2005. The NITE XML Toolkit meets the ICSI Meeting Corpus: Import, annotation, and browsing. In Samy Bengio and Hervé Bourlard, editors, *Machine Learning for Multimodal Interaction*, LNCS 3361, pages 111–121. Springer-Verlag, Berlin/Heidelberg.

Jean Carletta, Stefan Evert, Ulrich Heid, Jonathan Kilgour, Judy Robertson, and Holger Voormann. 2003. The NITE XML Toolkit: flexible annotation for multimodal language data. In *Behavior Research Methods, Instruments, and Computers*, special issue on Measuring Behavior, 35(3), pages 353–363.

Peter Wittenburg, Wim Peters, and Daan Broeder. 2002. Metadata proposals for corpora and lexica. In *LREC 2002 (3rd Int. Conf. on Language Resources and Evaluation)*, pages 1321–1326, Las Palmas.

Peter Wittenburg, Daan Broeder, and Paul Buitelaar. 2004. Towards metadata interoperability. In *NLPXML 2004 (4th Workshop on NLP and XML at ACL 2004)*, pages 9–16, Barcelona.

Exploration of Term Dependence in Sentence Retrieval

Keke Cai, Jiajun Bu, Chun Chen, Kangmiao Liu
College of Computer Science, Zhejiang University
Hangzhou, 310027, China
{caikeke,bjj,chenc,lkm}@zju.edu.cn

Abstract

This paper focuses on the exploration of term dependence in the application of sentence retrieval. The adjacent terms appearing in query are assumed to be related with each other. These assumed dependences among query terms will be further validated for each sentence and sentences, which present strong syntactic relationship among query terms, are considered more relevant. Experimental results have fully demonstrated the promising of the proposed models in improving sentence retrieval effectiveness.

1 Introduction

Sentence retrieval is to retrieve sentences in response to certain requirements. It has been widely applied in many tasks, such as passage retrieval (Salton et al, 1994), document summarization (Daumé and Marcu, 2006), question answering (Li, 2003) and novelty detection (Li and Croft 2005). A lot of different approaches have been proposed for this service, but most of them are based on term matching. Compared with document, sentence always consists of fewer terms. Limited information contained in sentence makes it quite difficult to implement such term based matching approaches.

Term dependence, which means that the presence or absence of one set of terms provides information about the probabilities of the presence or absence of another set of terms, has been widely accepted in recent studies of information retrieval. Taking into account the limited information about term distribution in sentence, the necessary of incorporating term dependence into sentence retrieval is clear.

Two kinds of dependence can be considered in the service of sentence retrieval. The first one occurs among query or sentence terms and another one occurs between query and sentence terms. This paper mainly focuses on the first kind of dependence and correspondingly proposes a new sentence retrieval model (TDSR). In general, TDSR model can be achieved through the following two steps:

The first step is to simulate the dependences among query terms and then represent query as a set of term combinations, terms of each of which are considered to be dependent with each other.

The second step is to measure the relevance of each sentence by considering the syntactic relationship of terms in each term combination formed above and then sort sentences according to their relevance to the given query.

The remainder is structured as follows: Section 2 introduces some related studies. Section 3 describes the proposed sentence retrieval model. In Section 4, the experimental results are presented and section 5 concludes the paper.

2 Related Works

Sentence retrieval is always treated as a special type of document retrieval (Larkey et al, 2002; Schiffman, 2002; Zhang et al, 2003). Weight function, such as tfidf algorithm, is used to construct the weighted term vectors of query and sentence. Similarity of these two vectors is then used as the evidence of sentence relevance. In fact, document retrieval differs from sentence retrieval in many ways. Thus, traditional docu-

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 97–100,
Prague, June 2007. ©2007 Association for Computational Linguistics

ment retrieval approaches, when implemented in the service of sentence retrieval, cannot achieve the expected retrieval performance.

Some systems try to utilize linguistic or other features of sentences to facilitate the detection of sentence relevance. In the study of White (2005), factors used for ranking sentences include the position of sentence in the source document, the words contained in sentence and the number of query terms contained in sentence. In another study (Collins-Thompson et al., 2002), semantic and lexical features are extracted from the initial retrieved sentences to filter out possible non-relevant sentences. Li and Croft (2005) chooses to describe a query by patterns that include both query words and required answer types. These patterns are then used to retrieve sentences.

Term dependence also has been tried in some sentence retrieval models. Most of these approaches realize it by referring to query expansion or relevance feedback. Terms that are semantically equivalent to the query terms or co-occurred with the query terms frequently can be selected as expanded terms (Schiffman, 2002). Moreover, query also can be expanded by using concept groups (Ohgaya et al., 2003). Sentences are then ranked by the cosine similarity between the expanded query vector and sentence vector. In (Zhang et al., 2003), blind relevance feedback and automatic sentence categorization based Support Vector Machine (SVM) are combined together to finish the task of sentence retrieval. In recent study, a translation model is proposed for monolingual sentence retrieval (Murdock and Croft, 2005). The basic idea is to use explicit relationships between terms to evaluate the translation probability between query and sentence. Although the translation makes an effective utilization of term relationships in the service of sentence retrieval, the most difficulty is how to construct the parallel corpus used for term translation.

Studies above have shown the positive effects of term dependence on sentence retrieval. However, it is considered that for the special task of sentence retrieval the potentialities of term dependence have not been fully explored. Sentence, being an integrated information unit, always has special syntactic structure. This kind of information is considered quite important to sentence relevance. How to incorporate this kind of information with information about dependences in query to realize the most efficient sentence retrieval is the main objective of this paper.

3 TDSR Model

As discussed above, the implementation of TDSR model consists of two steps. The following will give the detail description of each step.

3.1 Term Dependences in Query

Past studies have shown the importance of dependences among query terms and different approaches have been proposed to define the styles of term dependence in query. In this paper, the assumption of term dependence starts by considering the possible syntactic relationships of terms. For that the syntactic relationships can happen among any set of query terms, hence the assumption of dependence occurring among any query terms is considered more reasonable.

The dependences among all query terms will be defined in this paper. Based on this definition, the given query Q can be represented as: $Q = \{TS_1, TS_2, \ldots, TS_n\}$, each item of which contains one or more query terms. These assumed dependences will be further evaluated in each retrieved sentence and then used to define the relevance of sentence

3.2 Identification of Sentence Relevance

Term dependences defined above provide structure basis for sentence relevance estimate. However, their effects to sentence relevance identification are finally decided by the definition of sentence feature function. Sentence feature function is used to estimate the importance of the estimated dependences and then decides the relevance of each retrieved sentence.

In this paper, feature function is defined from the perspective of syntactic relationship of terms in sentence. The specific dependency grammar is used to describe such relationship in the form of dependency parse tree. A dependency syntactic relationship is an asymmetric relationship between a word called governor and another word called modifier. In this paper, MINIPAR is adopted as the dependency parser. An example of a dependency parse tree parsed by MINIPAR is shown in Figure 1, in which nodes are labeled by part of speeches and edges are labeled by relation types.

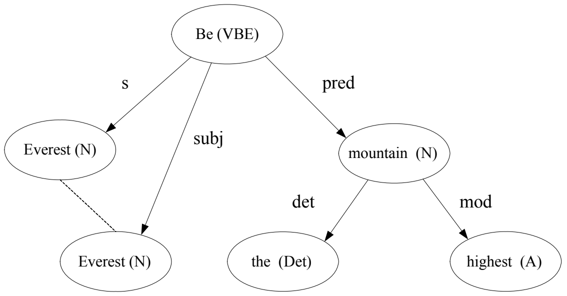


Figure 1. Dependency parse tree of sentence "Everest is the highest mountain".

As we know, terms within a sentence can be described by certain syntactic relationship (direct or indirect). Moreover, different syntactic relationships describe different degrees of associations. Given a query, the relevance of each sentence is considered different if query terms present different forms of syntactic relationships. This paper makes an investigation of syntactic relationships among terms and then proposes a novel feature function.

To evaluate the syntactic relationship of terms, the concept of association strength should be defined to each $TS_i \in Q$ with respect to each sentence S. It describes the association of terms in TS_i. The more closely they are related, the higher the value is. In this paper, the association strength of TS_i is valued from two aspects:

- Size of TS_i. Sentences containing more query terms are considered more relevant.
- Distance of TS_i. In the context of dependency parse tree, the link between two terms means their direct syntactic relationship. For terms with no direct linkage, their syntactic relationship can be described by the path between their corresponding nodes in tree. For example, in Figure 1 the syntactic relationship between terms "Everest" and "mountain" can be described by the path:

$$\text{Everest} \xrightarrow{\text{s}} \text{Be} \xrightarrow{\text{pred}} \text{mountain}$$

This paper uses term distance to evaluate terms syntactic relationship. Given two terms A and B, their distance $distance(A, B)$ is defined as the number of linkages between A and B with no consideration of direction. Furthermore, for the term set C, their distance is defined as:

$$D(C) = \frac{1}{N} * \sum_{q_i, q_j \in C} distance(q_i, q_j) \qquad (1)$$

where N is the number of term pairs of C.

Given the term set TS_i, the association strength of TS_i in sentence S is defined as:

$$AS(TS_i, S) = \alpha^{1/S(TS_i)} * \beta^{D(TS_i)} \qquad (2)$$

where $S(TS_i)$ is the size of term set TS_i and parameters α and β are valued between 0 and 1 and used to control the influence of each component on the computation of $AS(TS_i)$.

Based on the definition of association strength, the feature function of S can be further defined as:

$$F(S, Q) = \max_{TS_i \in Q} AS(TS_i, S) \qquad (3)$$

Taking the maximum association strength to evaluate sentence relevance conforms to the Disjunctive Relevance Decision principle (Kong et al., 2004). Based on the feature function defined above, sentences can be finally ranked according to the obtained maximum association strength.

4 Experiments

In this paper, the proposed method is evaluated on the data collection used in TREC novelty track 2003 and 2004 with the topics N1-N50 and N51-N100. Only the title portion of these TREC topics is considered.

To measure the performance of the suggested retrieval model, three traditional sentence retrieval models are also performed, i.e., TFIDF model (TFIDF), Okapi model (OKAPI) and KL-divergence model with Dirichlet smoothing (KLD). The result of TFIDF provides the baseline from which to compare other retrieval models.

Table 1 shows the non-interpolated average precision of each different retrieval models. The value in parentheses is the improvement over the baseline method. As shown in the table, TDSR model outperforms TFIDF model obviously. The improvements are respectively 15.3% and 10.2%.

	N1-N50	**N51-N100**
TFIDF	0.308	0.215
OKAPI	0.239 (-22.4)	0.165 (-23.3%)
KLD	0.281 (-8.8)	0.204 (-5.1%)
TDSR	0.355 (15.3%)	0.237 (10.2%)

Table 1. Average precision of each different retrieval models

Figure 2 and Figure 3 further depict the precision recall curve of each retrieval model when implemented on different query sets. The improvements of the proposed retrieval model indicated in these figures are clear. TDSR outperforms other retrieval models at any recall point.

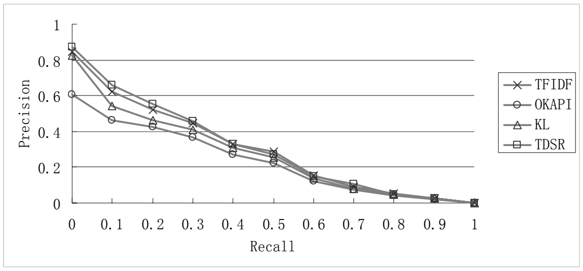


Figure 2. Precision-Recall Curve of Each Retrieval Model (N1-N50)

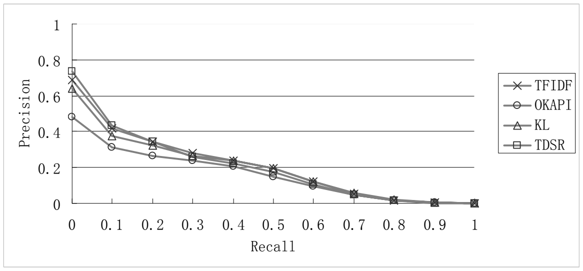


Figure 3. Precision-Recall Curve of Each Retrieval Model (N51-N100)

5 Conclusions

This paper presents a novel approach for sentence retrieval. Given a sentence, its relevance is measured by the degree of its support to the dependences between query terms. Term dependence, which has been widely considered in the studies of document retrieval, is the basis of this retrieval model. Experimental results show the promising of the proposed models in improving sentence retrieval performance.

References

Barry Schiffman. 2002. Experiments in Novelty Detection at Columbia University. In *Proceedings of the 11th Text REtrieval Conference*, pages 188-196.

Gerard Salton, James Allan, and Chris Buckley. 1994. Automatic structuring and retrieval of large text files. *Communication of the ACM*, 37(2): 97-108.

Hal Daumé III and Daniel Marcu. 2006. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 305-312, Sydney, Australia.

Kevyn Collins-Thompson, Paul Ogilvie, Yi Zhang, and Jamie Callan. 2002. Information filtering, Novelty detection, and named-page finding. In *Proceedings of the 11th Text REtrieval Conference*, National Institute of Standards and Technology.

Leah S. Larkey, James Allan, Margaret E. Connell, Alvaro Bolivar, and Courtney Wade. 2002. UMass at TREC 2002: Cross Language and Novelty Tracks. In *Proceeding of the Eleventh Text Retrieval Conference*, pages 721–732, Gaithersburg, Maryland.

Min Zhang, Chuan Lin, Yiqun Liu, Le Zhao, Liang Ma, and Shaoping Ma. 2003. THUIR at TREC 2003: Novelty, Robust, Web and HARD. In *Proceedings of 12th Text Retrieval Conference*, pages 137-148.

Ryen W. White, Joemon M. Jose, and Ian Ruthven. 2005. Using top-ranking sentences to facilitate effective information access. *Journal of the American Society for Information Science and Technology*, 56(10): 1113-1125.

Ryosuke Ohgaya, Akiyoshi Shimmura, Tomohiro Takagi, and Akiko N. Aizawa. 2003. Meiji University web and novelty track experiments at TREC 2003. In *Proceedings of the Twelfth Text Retrieval Conference*.

Vanessa Murdock and W. Bruce Croft. 2005. A translation Model for Sentence retrieval. HLT/EMNLP. In *Proceedings of the Conference on Human Language Technologies and Empirical Methods in Natural Language Processing*, pages 684-691.

Xiaoyan Li. 2003. Syntactic Features in Question Answering. In *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 455-456, Toronto, Canada.

Xiaoyan Li and W. Bruce Croft. 2005. Novelty detection based on sentence level patterns. In *Proceedings of ACM Fourteenth Conference on Information and Knowledge Management (CIKM)*, pages 744-751, Bremen, Germany.

Y.K. Kong, R.W.P. Luk, W. Lam, K.S. Ho and F.L. Chung. 2004. Passage-based retrieval based on parameterized fuzzy operators, *ACM SIGIR Workshop on Mathematical/Formal Methods for Information Retrieval*.

Minimum Bayes Risk Decoding for BLEU

Nicola Ehling and Richard Zens and Hermann Ney

Human Language Technology and Pattern Recognition
Lehrstuhl für Informatik 6 – Computer Science Department
RWTH Aachen University, D-52056 Aachen, Germany
{ehling,zens,ney}@cs.rwth-aachen.de

Abstract

We present a Minimum Bayes Risk (MBR) decoder for statistical machine translation. The approach aims to minimize the expected loss of translation errors with regard to the BLEU score. We show that MBR decoding on N-best lists leads to an improvement of translation quality.

We report the performance of the MBR decoder on four different tasks: the TC-STAR EPPS Spanish-English task 2006, the NIST Chinese-English task 2005 and the GALE Arabic-English and Chinese-English task 2006. The absolute improvement of the BLEU score is between 0.2% for the TC-STAR task and 1.1% for the GALE Chinese-English task.

1 Introduction

In recent years, statistical machine translation (SMT) systems have achieved substantial progress regarding their perfomance in international translation tasks (TC-STAR, NIST, GALE).

Statistical approaches to machine translation were proposed at the beginning of the nineties and found widespread use in the last years. The ”standard” version of the Bayes decision rule, which aims at a minimization of the sentence error rate is used in virtually all approaches to statistical machine translation. However, most translation systems are judged by their ability to minimize the error rate on the word level or n-gram level. Common error measures are the Word Error Rate (WER) and the Position Independent Word Error Rate (PER) as well as evaluation metric on the n-gram level like the BLEU and NIST score that measure precision and fluency of a given translation hypothesis.

The remaining part of this paper is structured as follows: after a short overview of related work in Sec. 2, we describe the MBR decoder in Sec. 3. We present the experimental results in Sec. 4 and conclude in Sec. 5.

2 Related Work

MBR decoder for automatic speech recognition (ASR) have been reported to yield improvement over the widely used maximum a-posteriori probability (MAP) decoder (Goel and Byrne, 2003; Mangu et al., 2000; Stolcke et al., 1997).

For MT, MBR decoding was introduced in (Kumar and Byrne, 2004). It was shown that MBR is preferable over MAP decoding for different evaluation criteria. Here, we focus on the performance of MBR decoding for the BLEU score on various translation tasks.

3 Implementation of Minimum Bayes Risk Decoding for the BLEU Score

3.1 Bayes Decision Rule

In statistical machine translation, we are given a source language sentence $f_1^J = f_1 \dots f_j \dots f_J$, which is to be translated into a target language sentence $e_1^I = e_1 \dots e_i \dots e_I$. Statistical decision theory tells us that among all possible target language sentences, we should choose the sentence which minimizes the Bayes risk:

$$\hat{e}_1^{\hat{I}} = \operatorname*{argmin}_{I, e_1^I} \left\{ \sum_{I', {e'}_1^{I'}} Pr({e'}_1^{I'} | f_1^J) \cdot L(e_1^I, {e'}_1^{I'}) \right\}$$

Here, $L(\cdot, \cdot)$ denotes the loss function under consideration. In the following, we will call this decision rule the MBR rule (Kumar and Byrne, 2004).

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 101–104,
Prague, June 2007. ©2007 Association for Computational Linguistics

Although it is well known that this decision rule is optimal, most SMT systems do *not* use it. The most common approach is to use the MAP decision rule. Thus, we select the hypothesis which maximizes the posterior probability $Pr(e_1^I|f_1^J)$:

$$\hat{e}_1^{\hat{I}} = \operatorname*{argmax}_{I,e_1^I} \left\{ Pr(e_1^I|f_1^J) \right\}$$

This decision rule is equivalent to the MBR criterion under a 0-1 loss function:

$$L_{0-1}(e_1^I, {e'}_1^{I'}) = \begin{cases} 1 & \text{if } e_1^I = {e'}_1^{I'} \\ 0 & \text{else} \end{cases}$$

Hence, the MAP decision rule is optimal for the sentence or string error rate. It is *not* necessarily optimal for other evaluation metrics as for example the BLEU score. One reason for the popularity of the MAP decision rule might be that, compared to the MBR rule, its computation is simpler.

3.2 Baseline System

The posterior probability $Pr(e_1^I|f_1^J)$ is modeled directly using a log-linear combination of several models (Och and Ney, 2002):

$$Pr(e_1^I|f_1^J) = \frac{\exp\left(\sum_{m=1}^{M} \lambda_m h_m(e_1^I, f_1^J)\right)}{\sum\limits_{I',{e'}_1^{I'}} \exp\left(\sum_{m=1}^{M} \lambda_m h_m({e'}_1^{I'}, f_1^J)\right)} \quad (1)$$

This approach is a generalization of the source-channel approach (Brown et al., 1990). It has the advantage that additional models $h(\cdot)$ can be easily integrated into the overall system.

The denominator represents a normalization factor that depends only on the source sentence f_1^J. Therefore, we can omit it in case of the MAP decision rule during the search process. Note that the denominator affects the results of the MBR decision rule and, thus, cannot be omitted in that case.

We use a state-of-the-art phrase-based translation system similar to (Matusov et al., 2006) including the following models: an n-gram language model, a phrase translation model and a word-based lexicon model. The latter two models are used for both directions: $p(f|e)$ and $p(e|f)$. Additionally, we use a word penalty, phrase penalty and a distortion penalty. The model scaling factors λ_1^M are optimized with respect to the BLEU score as described in (Och, 2003).

3.3 BLEU Score

The BLEU score (Papineni et al., 2002) measures the agreement between a hypothesis e_1^I generated by the MT system and a reference translation $\hat{e}_1^{\hat{I}}$. It is the geometric mean of n-gram precisions $\text{Prec}_n(\cdot,\cdot)$ in combination with a brevity penalty $\text{BP}(\cdot,\cdot)$ for too short translation hypotheses.

$$\begin{aligned} \text{BLEU}(e_1^I, \hat{e}_1^{\hat{I}}) &= \text{BP}(I, \hat{I}) \cdot \prod_{n=1}^{4} \text{Prec}_n(e_1^I, \hat{e}_1^{\hat{I}})^{1/4} \\ \text{BP}(I, \hat{I}) &= \begin{cases} 1 & \text{if } \hat{I} \geq I \\ \exp\left(1 - I/\hat{I}\right) & \text{if } \hat{I} < I \end{cases} \\ \text{Prec}_n(e_1^I, \hat{e}_1^{\hat{I}}) &= \frac{\sum\limits_{w_1^n} \min\{C(w_1^n|e_1^I), C(w_1^n|\hat{e}_1^{\hat{I}})\}}{\sum\limits_{w_1^n} C(w_1^n|e_1^I)} \end{aligned}$$

Here, $C(w_1^n|e_1^I)$ denotes the number of occurrences of an n-gram w_1^n in a sentence e_1^I. The denominator of the n-gram precisions evaluate to the number of n-grams in the hypothesis, i.e. $I - n + 1$.

As loss function for the MBR decoder, we use:

$$L[e_1^I, \hat{e}_1^{\hat{I}}] = 1 - \text{BLEU}(e_1^I, \hat{e}_1^{\hat{I}}) \,.$$

While the original BLEU score was intended to be used only for aggregate counts over a whole test set, we use the BLEU score at the sentence-level during the selection of the MBR hypotheses. Note that we will use this sentence-level BLEU score only during decoding. The translation results that we will report later are computed using the standard BLEU score.

3.4 Hypothesis Selection

We select the MBR hypothesis among the N best translation candidates of the MAP system. For each entry, we have to compute its expected BLEU score, i.e. the weighted sum over all entries in the N-best list. Therefore, finding the MBR hypothesis has a quadratic complexity in the size of the N-best list. To reduce this large work load, we stop the summation over the translation candidates as soon as the risk of the regarded hypothesis exceeds the current minimum risk, i.e. the risk of the current best hypothesis. Additionally, the hypotheses are processed according to the posterior probabilities. Thus, we can hope to find a good candidate soon. This allows for an early stopping of the computation for each of the remaining candidates.

3.5 Global Model Scaling Factor

During the translation process, the different submodels $h_m(\cdot)$ get different weights λ_m. These scaling factors are optimized with regard to a specific evaluation criteria, here: BLEU. This optimization describes the relation between the different models but does not define the absolute values for the scaling factors. Because search is performed using the maximum approximation, these absolute values are not needed during the translation process. In contrast to this, using the MBR decision rule, we perform a summation over all sentence probabilities contained in the N-best list. Therefore, we use a global scaling factor $\lambda_0 > 0$ to modify the individual scaling factors λ_m:

$$\lambda'_m = \lambda_0 \cdot \lambda_m \ \ , m = 1, ..., M.$$

For the MBR decision rule the modified scaling factors λ'_m are used instead of the original model scaling factors λ_m to compute the sentence probabilities as in Eq. 1. The global scaling factor λ_0 is tuned on the development set. Note that under the MAP decision rule any global scaling factor $\lambda_0 > 0$ yields the same result. Similar tests were reported by (Mangu et al., 2000; Goel and Byrne, 2003) for ASR.

4 Experimental Results

4.1 Corpus Statistics

We tested the MBR decoder on four translation tasks: the TC-STAR EPPS Spanish-English task of 2006, the NIST Chinese-English evaluation test set of 2005 and the GALE Arabic-English and Chinese-English evaluation test set of 2006. The TC-STAR EPPS corpus is a spoken language translation corpus containing the verbatim transcriptions of speeches of the European Parliament. The NIST Chinese-English test sets consists of news stories. The GALE project text track consists of two parts: newswire ("news") and newsgroups ("ng"). The newswire part is similar to the NIST task. The newsgroups part covers posts to electronic bulletin boards, Usenet newsgroups, discussion groups and similar forums.

The corpus statistics of the training corpora are shown in Tab. 1 to Tab. 3. To measure the translation quality, we use the BLEU score. With exception of the TC-STAR EPPS task, all scores are computed case-insensitive. As BLEU measures accuracy, higher scores are better.

Table 1: NIST Chinese-English: corpus statistics.

		Chinese	English
Train	Sentences	9 M	
	Words	232 M	250 M
	Vocabulary	238 K	412 K
NIST 02	Sentences	878	
	Words	26 431	24 352
NIST 05	Sentences	1 082	
	Words	34 908	36 027
GALE 06 news	Sentences	460	
	Words	9 979	11 493
GALE 06 ng	Sentences	461	
	Words	9 606	11 689

Table 2: TC-Star Spanish-English: corpus statistics.

		Spanish	English
Train	Sentences	1.2 M	
	Words	35 M	33 M
	Vocabulary	159 K	110 K
Dev	Sentences	1 452	
	Words	51 982	54 857
Test	Sentences	1 780	
	Words	56 515	58 295

4.2 Translation Results

The translation results for all tasks are presented in Tab. 4. For each translation task, we tested the decoder on N-best lists of size N=10 000, i.e. the 10 000 best translation candidates. Note that in some cases the list is smaller because the translation system did not produce more candidates. To analyze the improvement that can be gained through rescoring with MBR, we start from a system that has already been rescored with additional models like an n-gram language model, HMM, IBM-1 and IBM-4.

It turned out that the use of 1 000 best candidates for the MBR decoding is sufficient, and leads to exactly the same results as the use of 10 000 best lists. Similar experiences were reported by (Mangu et al., 2000; Stolcke et al., 1997) for ASR.

We observe that the improvement is larger for

Table 3: GALE Arabic-English: corpus statistics.

		Arabic	English
Train	Sentences	4 M	
	Words	125 M	124 M
	Vocabulary	421 K	337 K
news	Sentences	566	
	Words	14 160	15 320
ng	Sentences	615	
	Words	11 195	14 493

Table 4: Translation results BLEU [%] for the NIST task, GALE task and TC-STAR task (S-E: Spanish-English; C-E: Chinese-English; A-E: Arabic-English).

	TC-STAR S-E	NIST C-E		GALE A-E		GALE C-E	
decision rule	test	2002 (dev)	2005	news	ng	news	ng
MAP	52.6	32.8	31.2	23.6	12.2	14.6	9.4
MBR	52.8	33.3	31.9	24.2	13.3	15.4	10.5

Table 5: Translation examples for the GALE Arabic-English newswire task.

Reference	the saudi interior ministry announced in a report the implementation of the death penalty today, tuesday, in the area of medina (west) of a saudi citizen convicted of murdering a fellow citizen.
MAP-Hyp	saudi interior ministry in a statement to carry out the death sentence today in the area of medina (west) in saudi citizen *found guilty of killing* one of its citizens.
MBR-Hyp	*the* saudi interior ministry *announced* in a statement to carry out the death sentence today in the area of medina (west) in saudi citizen *was killed* one of its citizens.
Reference	faruq al-shar'a takes the constitutional oath of office before the syrian president
MAP-Hyp	farouk al-shara *leads sworn in by* the syrian president
MBR-Hyp	farouk al-shara *lead the constitutional oath before* the syrian president

low-scoring translations, as can be seen in the GALE task. For an ASR task, similar results were reported by (Stolcke et al., 1997).

Some translation examples for the GALE Arabic-English newswire task are shown in Tab. 5. The differences between the MAP and the MBR hypotheses are set in *italics*.

5 Conclusions

We have shown that Minimum Bayes Risk decoding on N-best lists improves the BLEU score considerably. The achieved results are promising. The improvements were consistent among several evaluation sets. Even if the improvement is sometimes small, e.g. TC-STAR, it is statistically significant: the absolute improvement of the BLEU score is between 0.2% for the TC-STAR task and 1.1% for the GALE Chinese-English task. Note, that MBR decoding is never worse than MAP decoding, and is therefore promising for SMT. It is easy to integrate and can improve even well-trained systems by tuning them for a particular evaluation criterion.

Acknowledgments

This material is partly based upon work supported by the Defense Advanced Research Projects Agency (DARPA) under Contract No. HR0011-06-C-0023, and was partly funded by the European Union under the integrated project TC-STAR (Technology and Corpora for Speech to Speech Translation, IST-2002-FP6-506738, http://www.tc-star.org).

References

P. F. Brown, J. Cocke, S. A. Della Pietra, V. J. Della Pietra, F. Jelinek, J. D. Lafferty, R. L. Mercer, and P. S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85, June.

V. Goel and W. Byrne. 2003. Minimum bayes-risk automatic speech recognition. *Pattern Recognition in Speech and Language Processing*.

S. Kumar and W. Byrne. 2004. Minimum bayes-risk decoding for statistical machine translation. In Proc. *Human Language Technology Conf. / North American Chapter of the Assoc. for Computational Linguistics Annual Meeting (HLT-NAACL)*, pages 169–176, Boston, MA, May.

L. Mangu, E. Brill, and A. Stolcke. 2000. Finding consensus in speech recognition: Word error minimization and other applications of confusion networks. *Computer, Speech and Language*, 14(4):373–400, October.

E. Matusov, R. Zens, D. Vilar, A. Mauser, M. Popović, S. Hasan, and H. Ney. 2006. The RWTH machine translation system. In Proc. *TC-Star Workshop on Speech-to-Speech Translation*, pages 31–36, Barcelona, Spain, June.

F. J. Och and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In Proc. *40th Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, pages 295–302, Philadelphia, PA, July.

F. J. Och. 2003. Minimum error rate training in statistical machine translation. In Proc. *41st Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, pages 160–167, Sapporo, Japan, July.

K. Papineni, S. Roukos, T. Ward, and W. J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proc. *40th Annual Meeting of the Assoc. for Computational Linguistics (ACL)*, pages 311–318, Philadelphia, PA, July.

A. Stolcke, Y. Konig, and M. Weintraub. 1997. Explicit word error minimization in N-best list rescoring. In Proc. *European Conf. on Speech Communication and Technology*, pages 163–166, Rhodes, Greece, September.

Disambiguating Between Generic and Referential *"You"* in Dialog[*]

Surabhi Gupta
Department of Computer Science
Stanford University
Stanford, CA 94305, US
surabhi@cs.stanford.edu

Matthew Purver
Center for the Study
of Language and Information
Stanford University
Stanford, CA 94305, US
mpurver@stanford.edu

Dan Jurafsky
Department of Linguistics
Stanford University
Stanford, CA 94305, US
jurafsky@stanford.edu

Abstract

We describe an algorithm for a novel task: disambiguating the pronoun *you* in conversation. *You* can be generic or referential; finding referential *you* is important for tasks such as addressee identification or extracting 'owners' of action items. Our classifier achieves 84% accuracy in two-person conversations; an initial study shows promising performance even on more complex multi-party meetings.

1 Introduction and Background

This paper describes an algorithm for disambiguating the generic and referential senses of the pronoun *you*.

Our overall aim is the extraction of *action items* from multi-party human-human conversations, concrete decisions in which one (or more) individuals take on a group commitment to perform a given task (Purver et al., 2006). Besides identifying the task itself, it is crucial to determine the *owner*, or person responsible. Occasionally, the name of the responsible party is mentioned explicitly. More usually, the owner is addressed directly and therefore referred to using a second-person pronoun, as in example (1).[1]

(1)
A: and um if **you can** get that binding point also maybe with a nice example that would be helpful for Johno and me.
B: Oh yeah uh O_K.

It can also be important to distinguish between singular and plural reference, as in example (2) where the task is assigned to more than one person:

(2)
A: So y- so **you guys will** send to the rest of us um a version of um, this, and - the - uh, description -
B: With sugge- yeah, suggested improvements and -

Use of *"you"* might therefore help us both in detecting the fact that a task is being assigned, and in identifying the owner. While there is an increasing body of work concerning *addressee identification* (Katzenmaier et al., 2004; Jovanovic et al., 2006), there is very little investigating the problem of *second-person pronoun resolution*, and it is this that we address here. Most cases of *"you"* do not in fact refer to the addressee but are generic, as in example (3); automatic referentiality classification is therefore very important.

(3)
B: Well, usually what **you** do is just wait until you think it's stopped,
and then **you** patch them up.

2 Related Work

Previous linguistic work has recognized that *"you"* is not always addressee-referring, differentiating between *generic* and *referential* uses (Holmes, 1998; Meyers, 1990) as well as idiomatic cases of *"you know"*. For example, (Jurafsky et al., 2002) found that *"you know"* covered 47% of cases, the referential class 22%, and the generic class 27%, with no significant differences in surface form (duration or vowel reduction) between the different cases.

While there seems to be no previous work investigating automatic classification, there is related work on classifying *"it"*, which also takes various referential and non-referential readings: (Müller, 2006) use lexical and syntactic features in a rule-based classifier to detect non-referential uses, achieving raw accuracies around 74-80% and F-scores 63-69%.

3 Data

We used the Switchboard corpus of two-party telephone conversations (Godfrey et al., 1992), and annotated the data with four classes: generic, referential singular, referential plural and a *reported referential* class, for mention in reported speech of an

[*] This work was supported by the CALO project (DARPA grant NBCH-D-03-0010) and ONR (MURI award N000140510388). The authors also thank John Niekrasz for annotating our test data.

[1] (1,2) are taken from the ICSI Meeting Corpus (Shriberg et al., 2004); (3,4) from Switchboard (Godfrey et al., 1992).

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 105–108,
Prague, June 2007. ©2007 Association for Computational Linguistics

	Training	Testing
Generic	360	79
Referential singular	287	92
Referential plural	17	3
Reported referential	5	1
Ambiguous	4	1
Total	673	176

Table 1: Number of cases found.

originally referential use (as the original addressee may not be the current addressee – see example (4)). We allowed a separate class for genuinely *ambiguous* cases. Switchboard explicitly tags *"you know"* when used as a discourse marker; as this (generic) case is common and seems trivial we removed it from our data.

(4)
B: Well, uh, I guess probably the last one I went to I met so many people that I had not seen in probably ten, over ten years.
It was like, don't **you** remember me.
And I am like no.
A: Am I related to **you**?

To test inter-annotator agreement, two people annotated 4 conversations, yielding 85 utterances containing *"you"*; the task was reported to be easy, and the kappa was 100%.

We then annotated a total of 42 conversations for training and 13 for testing. Different labelers annotated the training and test sets; none of the authors were involved in labeling the test set. Table 1 presents information about the number of instances of each of these classes found.

4 Features

All features used for classifier experiments were extracted from the Switchboard LDC Treebank 3 release, which includes transcripts, part of speech information using the Penn tagset (Marcus et al., 1994) and dialog act tags (Jurafsky et al., 1997). Features fell into four main categories:[2] *sentential* features which capture lexical features of the utterance itself; *part-of-speech* features which capture shallow syntactic patterns; *dialog act* features capturing the discourse function of the current utterance and surrounding context; and *context features* which give oracle information (i.e., the correct generic/referential label) about preceding uses of *"you"*. We also investigated using the presence of a question mark in the transcription as a feature, as a possible replacement for some dialog act features. Table 2 presents our features in detail.

N	Features
	Sentential Features (Sent)
2	you, you know, you guys
N	number of you, your, yourself
2	you (say|said|tell|told|mention(ed)|mean(t)|sound(ed))
2	you (hear|heard)
2	(do|does|did|have|has|had|are|could|should|n't) you
2	"if you"
2	(which|what|where|when|how) you
	Part of Speech Features (POS)
2	Comparative JJR tag
2	you (VB*)
2	(I|we) (VB*)
2	(PRP*) you
	Dialog Act Features (DA)
46	DA tag of current utterance i
46	DA tag of previous utterance $i-1$
46	DA tag of utterance $i-2$
2	Presence of any `question` DA tag (Q_DA)
2	Presence of `elaboration` DA tag
	Oracle Context Features (Ctxt)
3	Class of utterance $i-1$
3	Class of utterance $i-2$
3	Class of previous utterance by same speaker
3	Class of previous labeled utterance
	Other Features (QM)
2	Question mark

Table 2: Features investigated. N indicates the number of possible values (there are 46 DA tags; context features can be *generic*, *referential* or *N/A*).

5 Experiments and Results

As Table 1 shows, there are very few occurrences of the referential plural, reported referential and ambiguous classes. We therefore decided to model our problem as a two way classification task, predicting generic versus referential (collapsing referential singular and plural as one category). Note that we expect this to be the major useful distinction for our overall action-item detection task.

Baseline A simple baseline involves predicting the dominant class (in the test set, referential). This gives 54.59% accuracy (see Table 1).[3]

SVM Results We used LIBSVM (Chang and Lin, 2001), a support vector machine classifier trained using an RBF kernel. Table 3 presents results for

[2] Currently, features are all based on perfect transcriptions.

[3] Precision and recall are of course 54.59% and 100%.

Features	Accuracy	F-Score
Ctxt	45.66%	0%
Baseline	54.59%	70.63%
Sent	67.05%	57.14%
Sent + Ctxt + POS	67.05%	57.14%
Sent + Ctxt + POS + QM	76.30%	72.84%
Sent + Ctxt + POS + Q_DA	79.19%	77.50%
DA	80.92%	79.75%
Sent + Ctxt + POS + QM + DA	84.39%	84.21%

Table 3: SVM results: generic versus referential

various selected sets of features. The best set of features gave accuracy of 84.39% and f-score 84.21%.

Discussion Overall performance is respectable; precision was consistently high (94% for the highest-accuracy result). Perhaps surprisingly, none of the context or part-of-speech features were found to be useful; however, dialog act features proved very useful – using these features alone give us an accuracy of 80.92% – with the referential class strongly associated with question dialog acts.

We used manually produced dialog act tags, and automatic labeling accuracy with this fine-grained tagset will be low; we would therefore prefer to use more robust features if possible. We found that one such heuristic feature, the presence of question mark, cannot entirely substitute: accuracy is reduced to 76.3%. However, using only the binary Q_DA feature (which clusters together all the different kinds of question DAs) does better (79.19%). Although worse than performance with a full tagset, this gives hope that using a coarse-grained set of tags might allow reasonable results. As (Stolcke et al., 2000) report good accuracy (87%) for statement vs. question classification on manual Switchboard transcripts, such coarse-grained information might be reliably available.

Surprisingly, using the oracle context features (the correct classification for the previous *you*) alone performs worse than the baseline; and adding these features to sentential features gives no improvement. This suggests that the generic/referential status of each *you* may be independent of previous *you*s.

Features	Accuracy	F-Score
Prosodic only	46.66%	44.31%
Baseline	54.59%	70.63%
Sent + Ctxt + POS + QM + DA + Prosodic	84.39%	84.21%

Table 4: SVM results: prosodic features

Category	Referential	Generic
Count	294	340
Pitch (Hz)	156.18	143.98
Intensity (dB)	60.06	59.41
Duration (msec)	139.50	136.84

Table 5: Prosodic feature analysis

6 Prosodic Features

We next checked a set of prosodic features, testing the hypothesis that generics are prosodically reduced. Mean pitch, intensity and duration were extracted using Praat, both averaged over the entire utterance and just for the word "*you*". Classification results are shown in Table 4. Using only prosodic features performs below the baseline; including prosodic features with the best-performing feature set from Table 3 gives identical performance to that with lexical and contextual features alone.

To see why the prosodic features did not help, we examined the difference between the average pitch, intensity and duration for referential versus generic cases (Table 5). A one-sided t-test shows no significant differences between the average intensity and duration (confirming the results of (Jurafsky et al., 2002), who found no significant change in duration). The difference in the average pitch was found to be significant (p=0.2) – but not enough for this feature alone to cause an increase in overall accuracy.

7 Error Analysis

We performed an error analysis on our best classifier output on the training set; accuracy was 94.53%, giving a total of 36 errors.

Half of the errors (18 of 36) were ambiguous even for humans (the authors), if looking at the sentence alone without the neighboring context from the actual conversation – see (5a). Treating these examples thus needs a detailed model of dialog context.

The other major class of errors requires detailed knowledge about sentential semantics and/or the world – see e.g. (5b,c), which we can tell are referential because they predicate inter-personal comparison or communication.

In addition, as questions are such a useful feature (see above), the classifier tends to label all question cases as referential. However, generic uses do occur within questions (5d), especially if rhetorical (5e):

(5) a. so uh and if you don't have the money then use a credit card
b. I'm probably older than you
c. although uh I will personally tell you I used to work at a bank
d. Do they survive longer if you plant them in the winter time?
e. my question I guess are they really your peers?

8 Initial Multi-Party Experiments

The experiments above used two-person dialog data: we expect that multi-party data is more complex. We performed an initial exploratory study, applying the same classes and features to multi-party meetings.

Two annotators labeled one meeting from the AMI corpus (Carletta et al., 2006), giving a total of 52 utterances containing *"you"* on which to assess agreement: kappa was 87.18% for two way classification of generic versus referential. One of the authors then labeled a testing set of 203 utterances; 104 are generic and 99 referential, giving a baseline accuracy of 51.23% (and F-score of 67.65%).

We performed experiments for the same task: detecting generic versus referential uses. Due to the small amount of data, we trained the classifier on the Switchboard training set from section 3 (i.e. on two-party rather than multi-party data). Lacking part-of-speech or dialog act features (since the dialog act tagset differs from the Switchboard tagset), we used only the sentential, context and question mark features described in Table 2.

However, the classifier still achieves an accuracy of 73.89% and F-score of 74.15%, comparable to the results on Switchboard without dialog act features (accuracy 76.30%). Precision is lower, though (both precision and recall are 73-75%).

9 Conclusions

We have presented results on two person and multi-party data for the task of generic versus referential *"you"* detection. We have seen that the problem is a real one: in both datasets the distribution of the classes is approximately 50/50, and baseline accuracy is low. Classifier accuracy on two-party data is reasonable, and we see promising results on multi-party data with a basic set of features. We expect the accuracy to go up once we train and test on same-genre data and also add features that are more specific to multi-party data.

References

J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner. 2006. The AMI meeting corpus. In *MLMI 2005, Revised Selected Papers*.

C.-C. Chang and C.-J. Lin, 2001. *LIBSVM: a library for Support Vector Machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

J. J. Godfrey, E. Holliman, and J. McDaniel. 1992. SWITCHBOARD: Telephone speech corpus for research and development. In *Proceedings of IEEE ICASSP-92*.

J. Holmes. 1998. Generic pronouns in the Wellington corpus of spoken New Zealand English. *Kōtare*, 1(1).

N. Jovanovic, R. op den Akker, and A. Nijholt. 2006. Addressee identification in face-to-face meetings. In *Proceedings of the 11th Conference of the EACL*.

D. Jurafsky, E. Shriberg, and D. Biasca. 1997. Switchboard SWBD-DAMSL shallow-discourse-function annotation coders manual, draft 13. Technical Report 97-02, University of Colorado, Boulder.

D. Jurafsky, A. Bell, and C. Girand. 2002. The role of the lemma in form variation. In C. Gussenhoven and N. Warner, editors, *Papers in Laboratory Phonology VII*, pages 1–34.

M. Katzenmaier, R. Stiefelhagen, and T. Schultz. 2004. Identifying the addressee in human-human-robot interactions based on head pose and speech. In *Proceedings of the 6th International Conference on Multimodal Interfaces*.

M. Marcus, G. Kim, M. Marcinkiewicz, R. MacIntyre, A. Bies, M. Ferguson, K. Katz, and B. Schasberger. 1994. The Penn treebank: Annotating predicate argument structure. In *ARPA Human Language Technology Workshop*.

M. W. Meyers. 1990. Current generic pronoun usage. *American Speech*, 65(3):228–237.

C. Müller. 2006. Automatic detection of nonreferential *It* in spoken multi-party dialog. In *Proceedings of the 11th Conference of the EACL*.

M. Purver, P. Ehlen, and J. Niekrasz. 2006. Detecting action items in multi-party meetings: Annotation and initial experiments. In *MLMI 2006, Revised Selected Papers*.

E. Shriberg, R. Dhillon, S. Bhagat, J. Ang, and H. Carvey. 2004. The ICSI Meeting Recorder Dialog Act (MRDA) Corpus. In *Proceedings of the 5th SIGdial Workshop*.

A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, C. V. Ess-Dykema, R. Martin, and M. Meteer. 2000. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–373.

On the formalization of Invariant Mappings for Metaphor Interpretation

Rodrigo Agerri, John Barnden, Mark Lee and Alan Wallington
School of Computer Science, Univ. of Birmingham
B15 2TT Birmingham, UK
r.agerri@cs.bham.ac.uk

Abstract

In this paper we provide a formalization of a set of default rules that we claim are required for the transfer of information such as causation, event rate and duration in the interpretation of metaphor. Such rules are domain-independent and are identified as invariant adjuncts to any conceptual metaphor. We also show a way of embedding the invariant mappings in a semantic framework.

1 Introduction

It is generally accepted that much of everyday language shows evidence of metaphor. We assume the general view that metaphor understanding involves some notion of events, properties, relations, etc. that are transferred from the source domain into the target domain. In this view, a metaphorical utterance conveys information about the target domain. We are particularly interested in the metaphorical utterances that we call *map-transcending*. Consider the following example:

(1) "McEnroe starved Connors to death."

We do not address in this paper the issue of when an utterance is to be considered metaphorical. Instead, we aim to offer an explanation of how a metaphorical utterance such as (1) can be interpreted. If we infer, using our knowledge about McEnroe and Connors, that (1) is used to describe a tennis match, it can be understood as an example of the conceptual metaphors (or, in our terminology, 'metaphorical views') DEFEAT AS DEATH and NECESSITIES AS FOOD. However, these metaphorical views would not contain any relationship that maps the specific *manner* of dying that constitutes *being starved to death* (we say that "starving" is a map-transcending entity). Yet one could argue that the *manner* of Connors's death is a crucial part of the informational contribution of (1).

A possible solution would be to create a new view-specific mapping that goes from the form of killing involved in *starving to death* to some process in sport, but such enrichment of mappings would be needed for many other verbs or verbal phrases that refer to other *ways* in which death is brought about, each requiring a specific specific mapping when occurring in a metaphorical utterance. Thus, finding adequate mappings could become an endless and computational intensive process. Moreover, there are even cases in which we may not find a plausible mapping. Consider the following description of the progress of a love affair:

(2) "We're spinning our wheels."

It is not very clear what could be a target correspondent for 'wheels'. We have developed an AI system called ATT-Meta for metaphor interpretation (Barnden et al., 2002) that employs reasoning within the terms of the source domain using various sources of information including *world* and *linguistic knowledge*. The reasoning connects unmapped ideas used by utterances, such as wheels and starving, to other source-domain ideas for which a mapping is already known. These known mappings may be constituents of particular metaphorical view, but previous work (Barnden et al., 2003; Wallington et al., 2006) has

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 109–112,
Prague, June 2007. ©2007 Association for Computational Linguistics

shown evidence that there are metaphorical aspects (such as causal relations between events) that, subject to being called, invariantly map from source to target (we call these mappings View-Neutral Mapping Adjuncts or VNMAs) irrespective of whatever specific metaphorical views are in play. These allow many mapping effects, which would otherwise have to be duplicated across all view-specific mappings, to be factored out into separate mappings. In our approach, source domain reasoning takes place in a special, protected computational context that we call the "pretence space". We use the term 'reality' to refer to the space outside the pretence where propositions are about reality as the understander sees it.

Currently ATT-Meta implements the VNMAs by including them in view-specific rules, but we plan to make the system more modular and its view-specific mappings more economical by implementing VNMAs as separate default rules. The first step towards that goal is to provide a formalization of these mappings and to show their role in metaphor interpretation. In order to do so, we provide a semantic representation of how these VNMAs work by adopting Segmented Discourse Representation Theory (Asher and Lascarides, 2003) to capture the main aspects of the ATT-Meta approach.

2 Knowledge and Inference

If (1) is being used metaphorically to describe the result of a tennis match, a plausible target interpretation would be that McEnroe defeated Connors in a slow manner by performing some actions to deprive him of his usual playing style. Assuming a commonsensical view of the world, a within-pretence meaning would be that McEnroe starved Connors to death in the real, biological sense. The inferencing within the pretence can then conclude that McEnroe *caused* Connors's death by *depriving* or disabling him. Leaving some details aside, the partial logical form (in the pretence) of the metaphorical utterance (1) may be represented as follows (without taking into account temporal issues):

(i) $\exists x, y, e(McEnroe(x) \wedge Connors(y)$
$\wedge starve - to - death(e, x, y))$

This says that there is an event e of x starving y to death (we also use the notion of event to describe situations, processes, states, etc.). It may be suggested that if we were trying to map the partial expression (i), its correspondent proposition in the target could be expressed by this formula:

(ii) $\exists x, y, e(McEnroe(x) \wedge Connors(y) \wedge$
$defeat(e, x, y))$

According to this, the event of x defeating y in the reality would correspond to the event of x starving y to death in the pretence. However, by saying "McEnroe starved Connors to death" instead of simply "McEnroe killed Connors" the speaker is not merely intending to convey that McEnroe defeated Connors, but rather something related to the manner in which Connors was defeated. Following this, *starving* may be decomposed into the cause e_1 and its effect, namely, "being deprived of food":

(iii) $\exists x, y, z, e_1, e_2, e_3(McEnroe(x) \wedge$
$Connors(y) \wedge food(z) \wedge starve(e_1, x, y) \wedge$
$death(e_2, y) \wedge deprived(e_3, y, z) \wedge$
$cause(e_1, e_3))$

Now, by means of lexical information regarding "starving", it can be inferred that McEnroe deprived Connors of a necessity (see, e.g., Wordnet), namely, of the food required for his normal functioning (the NECESSITIES AS FOOD metaphorical view would provide mappings to transfer food to the type of shots that Connors *needs* to play his normal game). In other words, Connors is defeated by the particular means of depriving him of a necessity (food) which means that being deprived causes Connors's defeat. This fits well with the interpretation of (1) where McEnroe's playing deprived Connors of his usual game. Moreover, linguistic knowledge also provides the fact that starving someone to death is a gradual, slow process. The result of within-pretence inferencing may be represented as follows:

(iv) $\exists x, y, z, e_1, e_2, e_3(McEnroe(x) \wedge$
$Connors(y) \wedge food(z) \wedge starve(e_1, x, y) \wedge$
$death(e_2, y) \wedge deprived(e_3, y, z) \wedge$
$cause(e_1, e_3) \wedge cause(e_3, e_2) \wedge rate(e_1, slow))$

'Slow' refers to a commonsensical concept in the pretence related to the progress rate of *starving*. Now, the existing mapping DEFEAT AS DEATH can be applied to derive, outside the pretence, that McEnroe defeated Connors, but no correspondences are available to account for the fact that McEnroe *caused* the defeat of Connors by depriving him of his normal play. We appear to have a problem also to map the slow progress *rate* of a process like starving.

3 VNMAs in a Semantic Framework

In the ATT-Meta approach to metaphor interpretation, the mappings of *caused* and *rate* discussed above are accomplished by a type of default mappings that we specify as VNMAs (the Causation and Rate VNMAs, respectively; see (Wallington and Barnden, 2006) for an informal but detailed description of a number of VNMAs). The idea is that there are relationships and properties (causation, rate, etc.) between two events or entities that identically transfer from the pretence to the reality. We use the $\mapsto$ symbol to express that this mapping is a default. The VNMAs involved in the interpretation of (1) can be represented as follows:

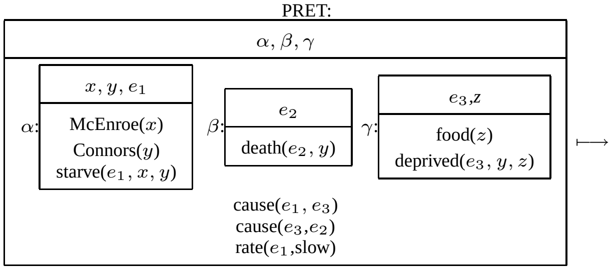
Causation: $\forall e_1, e_2(cause(e_1, e_2)_{pret} \mapsto cause(e_1, e_2)_{rlt})$

The Rate VNMA transfers the qualitative rate of progress of events in the source domain to the qualitative rate of progress of its mappee:

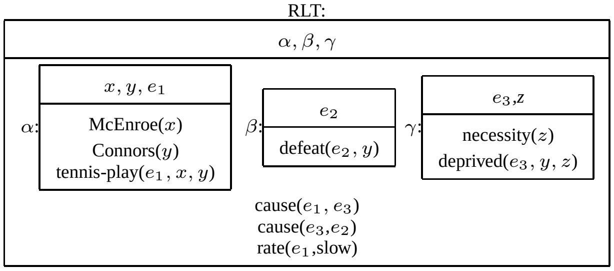
Rate: $\forall e, r(rate(e, r)_{pret} \mapsto rate(e, r)_{rlt})$

Embedding the VNMAs in a semantic framework for metaphor interpretation is useful as a first step towards their implementation as default rules in the ATT-Meta system, but it is also interesting in its own right to show the contribution that the ATT-Meta approach can make towards the semantics of metaphor. In the somewhat simplified discussion on the within-pretence reasoning and mappings necessary to interpret metaphorical utterances such as (1), we have been using various sources of information that interact in the processing of the utterance: a) View-specific mappings provided by the relevant metaphorical views (DEFEAT AS DEATH and NECESSITIES AS FOOD); b) Linguistic and contextual information necessary for reasoning in the pretence; c) Relations and properties between events such as *causation* and *rate* that are inferred in the pretence; d) VNMAs that transfer within-pretence event relations and properties to reality.

There are two prominent computationally-oriented semantic approaches (Hobbs, 1996) and (Asher and Lascarides, 2003) that take into account contextual and linguistic information and stress the importance of relations between text segments in discourse interpretation. In fact, the incorporation of the above types of information ties in well with the SDRT (Asher and Lascarides, 2003) view of language understanding. For example, we can think of the pretence space as a Segmented Discourse Representation Structure (SDRS) representing the result of within-pretence inference which can be mapped by using various view-specific and invariant mappings to reality. In other words, we can see the pretence SDRS as the input for what the ATT-Meta system does when interpreting metaphor – it will reason with it, producing an output of inferred reality facts which we may also represent by means of an SDRS. The result of reasoning in the pretence to interpret (1) would now looks as follows:



where α and β are labels for DRSs representing events, PRET for a pretence space and $\longmapsto$ mappings (VNMAs and central mappings) needed in the interpretation of the metaphorical utterance. Importantly, the VNMAs would pick upon aspects such as causation and rate from pretence to transfer them to reality producing an output which could also be represented as a SDRS:



Note that this formal representation integrates the systematicity of mapping invariantly certain aspects of metaphorical utterances by formulating them as relations between events that can be represented as relations and properties of DRSs. For this purpose we need to modify the construction rules of SDRSs to be able to infer properties and relations involving individuals and not only DRSs' labels. In addition to this, we have shown in the previous section how ATT-Meta source domain reasoning captures the interaction of the various sources of knowledge used to infer causation and rate in the pretence. Furthermore, studying the interaction between VNMAs and discourse relations may allow us to extend the study of metaphor to discourse.

4 Concluding Remarks

Following the ATT-Meta claim metaphors often convey crucial information via VNMAs, we can reanalyze example (1) so that the effects of the NECESSITIES AS FOOD mapping are obtained by VNMAs. In the pretence, the food is something Connors needs for proper functioning: i.e., it is necessary that Connors have the food in order to function properly. The necessity here is covered by the Modality VNMA, which maps relative degrees of necessity, possibility, obligation, etc., from pretence to reality. Moreover, the functioning properly would be covered by the Function and Value-Judgement (levels of goodness, importance, etc. map identically to levels of goodness, etc.). So all that is left is the possession which could be covered by a STATE AS POSSESSION mapping.

Formal semantic approaches (Asher and Lascarides, 2003) do not account for metaphorical utterances including map-transcending entities. Other works (Carbonell, 1982; Hobbs, 1990; Martin, 1990; Narayanan, 1997) have addressed source domain reasoning to a limited extent, but its role in metaphor interpretation has not previously been adequately investigated. Moreover, map-transcending entities pose a problem for analogy-based approaches to metaphor interpretation (Falkenhainer et al., 1989), which require the discovery of an elaborate structural similarity between the source and target domains and/or the imposition of unmapped source domain structures on the target domain, whereas part of our approach is that the unmapped source domain structure introduced by the utterance is by default not carried over.

Acknowledgements Supported by EPSRC EP/C538943/1 and GR/M64208 grants.

References

Nicholas Asher and Alex Lascarides. 2001. The semantics and pragmatics of metaphor. In P. Bouillon and F. Busa, editors, *The Language of Word Meaning*, pages 262–289. Cambridge University Press.

Nicholas Asher and Alex Lascarides. 2003. *Logics of Conversation*. Cambridge University Press.

John Barnden, Sheila Glasbey, Mark Lee, and Alan Wallington. 2002. Reasoning in metaphor understanding: The att-meta approach and system. In *19th Conference on Computational Linguistics (COLING-2002)*.

John Barnden, Sheila Glasbey, Mark Lee, and Alan Wallington. 2003. Domain-transcending mappings in a system for metaphorical reasoning. In *Conference Companion to the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2003)*, pages 57–61.

Jaime Carbonell. 1982. Metaphor: An inescapable phenomenon in natural-language comprehension. In W. Lehnert and M. Ringle, editors, *Strategies for Natural Language Processing*, pages 415–434. Lawrence Erlbaum, Hillsdale, NJ.

BrianFalkenhainer, Kenneth Forbus, and Dedre Gentner. 1989. The structure-mapping engine: algorithm and examples. *Artificial Intelligence*, 41(1):1–63.

Jerry Hobbs. 1990. *Literature and Cognition*. CSLI, Lecture Notes, Stanford.

Jerry Hobbs. 1996. An approach to the structure of discourse. In D. Everett, editor, *Discourse: Linguistic, Computational and Philosophical Perspectives*.

James Martin. 1990. *A computational model of metaphor interpretation*. Academic Press, New York.

Srini Narayanan. 1997. *KARMA: Knowledge-based action representations for metaphor and aspect*. Ph.D. thesis, Computer Science Division, EECS Department, University of California, Berkeley, August.

Alan Wallington and John Barnden. 2006. Similarity as a basis for metaphor: Invariant transfer and the role of VNMAs. Technical Report CSRP-06-02, School of Computer Science, Univ. of Birmingham, December.

Alan Wallington, John Barnden, Sheila Glasbey, and Mark Lee. 2006. Metaphorical reasoning with an economical set of mappings. *Delta*, 22(1).

Real-Time Correction of Closed-Captions

P. Cardinal, G. Boulianne, M. Comeau, M. Boisvert
Centre de recherche Informatique de Montreal (CRIM)
Montreal, Canada
patrick.cardinal@crim.ca

Abstract

Live closed-captions for deaf and hard of hearing audiences are currently produced by stenographers, or by voice writers using speech recognition. Both techniques can produce captions with errors. We are currently developing a correction module that allows a user to intercept the real-time caption stream and correct it before it is broadcast. We report results of preliminary experiments on correction rate and actual user performance using a prototype correction module connected to the output of a speech recognition captioning system.

1 Introduction

CRIM's automatic speech recognition system has been applied to live closed-captioning of french-canadian television programs (Boulianne et al., 2006). The low error rate of our approach depends notably on the integration of the re-speak method (Imai et al., 2002) for a controlled acoustic environment, automatic speaker adaptation and dynamic updates of language models and vocabularies, and was deemed acceptable by several Canadian broadcasters (RDS,CPAC,GTVA and TQS) who have adopted it over the past few years for captioning sports, public affairs and newscasts.

However, for sensitive applications where error rates must practically be zero, or other situations where speech recognition error rates are too high, we are currently developing a real-time correction interface. In essence, this interface allows a user to correct the word stream from speech recognition before it arrives at the closed-caption encoder.

2 Background

Real-time correction must be done within difficult constraints : with typical captioning rates of 130 words per minute, and 5 to 10% word error rate, the user must correct between 6 and 13 errors per minute. In addition, the process should not introduce more than a few seconds of additional delay over the 3 seconds already needed by speech recognition.

In a previous work, (Wald et al., 2006) explored how different input modalities, such as mouse/keyboard combination, keyboard only or function keys to select words for editing, could reduce the amount of time required for correction. In (Bateman et al., 2000), the correction interface consisted in a scrolling window which can be edited by the user using a text editor style interface. They introduced the idea of a controllable delay during which the text can be edited.

Our approach combines characteristics of the two previous systems. We use a delay parameter, which can be modified online, for controlling the output rate. We also use the standard mouse/keyboard combination for selecting and editing words. However we added, for each word, a list of alternate words that can be selected by a simple mouse click; this simplifies the edition process and speeds up the correction time. However, manual word edition is still available.

Another distinctive feature of our approach is the fixed word position. When a word appears on screen, it will remain in its position until it is sent

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 113–116,
Prague, June 2007.

out. This allows the user to focus on the words and not be distracted by word-scrolling or any other word movement.

3 Correction Software

The correction software allows edition of the closed-captions by intercepting them while they are being sent to the encoder. Both assisted and manual corrections can be applied to the word stream.

Assisted correction reduces the number of operations by presenting a list of alternate words, so that a correction can be done with a simple mouse click. Manual correction requires editing the word to be changed and is more expensive in terms of delay. As a consequence, the number of these operations should be reduced to a strict minimum.

The user interface shown in figure 1 has been designed with this consideration in mind. The principal characteristic of the interface is that there is no scrolling. Words never move; instead the matrix is filled from left to right, top to bottom, with words coming from the speech recognition, in synchronisation with the audio. When the bottom right of the matrix is reached, filling in starts from the upper left corner again. Words appear in blue while they are editable, and in red once they have been sent to the caption encoder. Thus a blue ”window”, corresponding to the interval during which words can be edited, moves across the word matrix, while the words themselves remain fixed.

For assisted correction, the list of available alternatives is presented in a list box under each word. These lists are always present, instead of being presented only upon selection of a word. In this way the user has the opportunity of scanning the lists in advance whenever his time budget allows.

The selected word can also be deleted with a single click. Different shortcut corrections, as suggested in (Wald et al., 2006) can also be applied depending on the mouse button used to select the word: a left button click changes the gender (masculin or feminin) of the word while a right button click changes the plurality (singular or plural) of the word. These available choices are in principle excluded from the list box choices.

To apply a manual correction, the user simply clicks the word with the middle button to make it editable; modifications are done using the keyboard.

Two users can run two correction interfaces in parallel, on alternating sentences. This configuration avoids the accumulation of delays. This functionality may prove useful if the word rate is so high that it becomes too difficult to keep track of the word flow. In this mode, the second user can begin the correction of a new sentence even if the first has not yet completed the correction of his/her sentence. Only one out of two sentences is editable by each user. The synchronisation is on a sentence basis.

3.1 Alternate word lists

As described in the previous section, the gender/plurality forms of the word are implicitly included and accessible through a simple left/right mouse click. Other available forms explicitly appear in a list box. This approach has two major benefits. First, when a gender/plurality error is detected by the user, no delay is incurred from scanning the choices in the list box. Second, since the gender/plurality forms are not included in the list box, their place becomes available for additional alternate words.

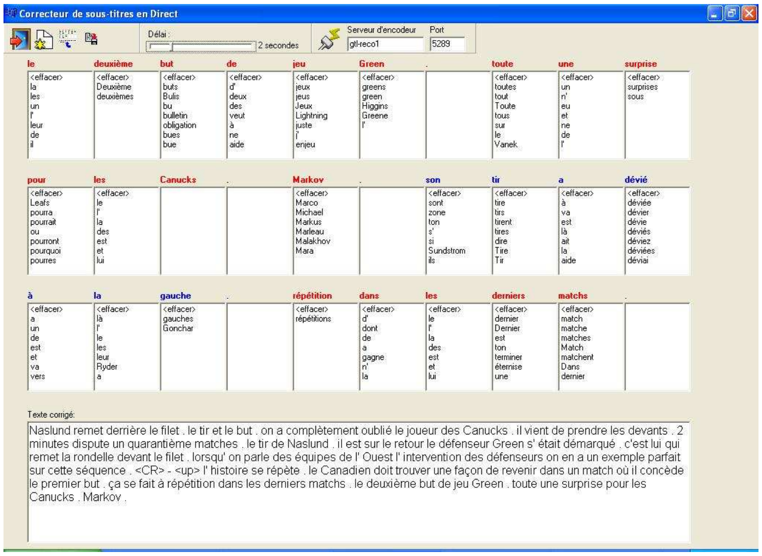
The main problem is to establish word lists short enough to reduce scanning time, but long enough to contain the correct form. For a given word output by the speech recognition system, the alternate words should be those that are most likely to be confused by the recognizer.

We experimented with two pre-computed sources of alternate word lists:

1. A list of frequently confused words was computed from all the available closed-captions of our speech recognition system for which corresponding exact transcriptions exist. The training and development sets were made up of 1.37M words and 0.17M words, respectively.

2. A phoneme based confusion matrix was used for scoring the alignment of each word of the vocabulary with every other word of the same vocabulary. The alignment program was an implementation of the standard dynamic programming technique for string alignment (Cormen et al., 2001).

Each of these techniques yields a list of alternate words with probabilities based on substitution like-



Délai : 2 secondes

Serveur d'encodeur gtl-reco1

Port 5289

le	deuxième	but	de	jeu	Green	.	toute	une	surprise
<effacer> la les un l' leur de il	<effacer> Deuxième deuxièmes	<effacer> buts Bulis bu bulletin obligation bues bue	<effacer> d' deux des veut à ne aide	<effacer> jeux jeus Jeux Lightning juste j' enjeu	<effacer> greens green Higgins Greene l'		<effacer> toutes tout Toute tous sur le Vanek	<effacer> un n' eu et ne de l'	<effacer> surprises sous

pour	les	Canucks	.	Markov	.	son	tir	a	dévié
<effacer> Leafs pourra pourrait ou pourront pourquoi pourres	<effacer> le l' la des est et lui			<effacer> Marco Michael Markus Marleau Malakhov Mara		<effacer> sont zone ton s' si Sundstrom ils	<effacer> tire tirs tirent tires dire Tire Tir	<effacer> à va est là ait la aide	<effacer> déviée dévier dévie déviés déviez déviées déviai

à	la	gauche	.	répétition	dans	les	derniers	matchs	.
<effacer> a un de est et va vers	<effacer> là l' le les leur Ryder a	<effacer> gauches Gonchar		<effacer> répétitions	<effacer> d' dont de a gagne n' la	<effacer> le l' la des est et lui	<effacer> dernier Dernier est ton terminer éternise une	<effacer> match matche matches Match matchent Dans dernier	

Texte corrigé:

Naslund remet derrière le filet . le tir et le but . on a complètement oublié le joueur des Canucks . il vient de prendre les devants . 2 minutes dispute un quarantième matches . le tir de Naslund . il est sur le retour le défenseur Green s' était démarqué . c'est lui qui remet la rondelle devant le filet . lorsqu' on parle des équipes de l' Ouest l' intervention des défenseurs on en a un exemple parfait sur cette séquence . <CR> - <up> l' histoire se répète . le Canadien doit trouver une façon de revenir dans un match où il concède le premier but . ça se fait à répétition dans les derniers matchs . le deuxième but de jeu Green . toute une surprise pour les Canucks . Markov .

Figure 1: Real-time corrector software.

Source of alternates	coverage (%)
Word confusion matrix	52%
Phoneme confusion matrix	37%
Combined	60%

Table 1: *Coverage of substitutions (dev set).*

lihoods. Table 1 shows how many times substitutions in the development set could be corrected with a word in the list, for each list and their combination.

To combine both lists, we take this coverage into consideration and the fact that 48% of the words were common to both lists. On this basis, we have constructed an alternate list of 10 words comprised of the most likely 7 words of case 1; the remaining 3 words are the most probable substitutions from the remaining words of both lists.

3.2 Real-time List Update

The previous technique can only handle simple substitutions: a word that is replaced by another one. Another frequent error in speech recognition is the replacement of a single word by several smaller ones. In this case, the sequence of errors contains one substitution and one or more insertions. From the interface point of view, the user must delete some words before editing the last word in the sequence.

To assist the user in this case, we have implemented the following procedure. When a word is deleted by the user, the phonemes of this word are concatenated with those of the following words. The resulting sequence of phonemes is used to search the dictionary for the most likely words according to the pronunciation. These words are dynamically added to the list appearing under the preceding word. The search technique used is the same alignment procedure implemented for computing the confusion matrix based on phoneme confusion.

4 Results

In this section we present the results of two preliminary experiments. In the first one, we simulated a perfect correction, as if the user had an infinite amount of time, to determine the best possible results that can be expected from the alternate word lists. In the second experiment, we submitted a prototype to users and collected performance measurements.

4.1 Simulation Results

The simulation is applied to a test set consisting of a 30 minute hockey game description for which closed-captions and exact transcripts are available. We aligned the produced closed-captions with their corrected transcripts and replaced any incorrect word by its correct counterpart if it appeared in the alternate list. In addition, all insertion errors were deleted. Table 2 shows the word error rate (WER)

Source of alternates	WER
Original closed-captions	5.8%
Phoneme confusion matrix	4.4%
Word confusion matrix	3.1%
Combined	2.9%

Table 2: *Error rate for perfect correction.*

	Delay	
	2 seconds	15 seconds
test duration	30 minutes	8 minutes
# of words	4631	1303
# of editions	21	28
WER before	6.8%	6.2%
WER after	6.1%	2.5%
Gain (relative %)	8.1%	58.7%

Table 3: *Error rate after user correction.*

obtained for different alternate word lists.

The word confusion matrix captures most of the substitutions. This behavior was expected since the matrix has been trained explicitely for that purpose. The performance should increase in the future as the amount of training data grows. In comparison, the contribution of words from the phoneme confusion matrix is clearly limited.

The corrected word was the first in the list 35% of the time, while it was in the first three 59% of the time. We also simulated the effect of collapsing words in insertion-substitution sequences to allow corrections of insertions : the increase in performance was less than 0.5%.

4.2 User Tests

Experiments were performed by 3 unacquainted users of the system on hockey game descriptions. In one case, we allowed a delay of 15 seconds; the second case allowed a 2 second delay to give a preliminary assessment of user behavior in the case of minimum-delay real-time closed-captioning. Table 3 shows the error rate before and after correction.

The results show that a significant WER decrease is achieved by correcting using a delay of 15 seconds. The reduction with a 2 second delay is minor; with appropriate training, however, we can expect the users to outperform these preliminary results.

5 Conclusion and Future Work

We are currently developing a user interface for correcting live closed-captions in real-time. The interface presents a list of alternatives for each automatically generated word. The theoretical results that assumes the user always chooses the correct suggested word shows the potential for large error reductions, with a minimum of interaction. When larger delays are allowed, manual edition of words for which there is no acceptable suggested alternative can yield further improvements.

We tested the application for real-time text correction produced in a real-world application. With users having no prior experience and with only a 15 second delay, the WER dropped from 6.1% to 2.5%.

In the future, users will be trained on the system and we expect an important improvement in both accuracy and required delay. We will also experiment the effect of running 2 corrections in parallel for more difficult tasks. Future work also includes the integration of an automatic correction tool for improving or highlighting the alternate word list.

References

A. Bateman, J. Hewitt, A. Ariyaeeinia, P. Sivakumaran, and A. Lambourne. 2000. *The Quest for The Last 5%: Interfaces for Correcting Real-Time Speech-Generated Subtitles* Proceedings of the 2000 Conference on Human Factors in Computing Systems (CHI 2000), April 1-6, The Hague, Netherlands.

T.H. Cormen, C.E. Leiserson, R.L. Rivest, and C. Stein 2001. *Introduction to Algorithms* second edition, MIT Press, Cambridge, MA.

G. Boulianne, J.-F. Beaumont, M. Boisvert, J. Brousseau, P. Cardinal, C. Chapdelaine, M.Comeau, P. Ouellet, and F. Osterrath. 2006. *Computer-assisted closed-captioning of live TV broadcasts in French* Proceedings of the 2006 Interspeech - ICSLP, September 17-21, Pittsburg, US.

T. Imai, A. Matsui, S. Homma, T. Kobayakawa, O. Kazuo, S. Sato, and A. Ando 2002. *Speech Recognition with a respeak method for subtiling live broadcast* Proceedings of the 2002 ICSLP, September 16-20, Orlando, US.

Wald, M. 2006 *Creating Accessible Educational Multimedia through Editing Automatic Speech Recognition Captioning in Real Time.* International Journal of Interactive Technology and Smart Education : Smarter Use of Technology in Education 3(2) pp. 131-142

Learning to Rank Definitions to Generate Quizzes for Interactive Information Presentation

Ryuichiro Higashinaka and **Kohji Dohsaka** and **Hideki Isozaki**
NTT Communication Science Laboratories, NTT Corporation
2-4, Hikaridai, Seika-cho, Kyoto 619-0237, Japan
{rh,dohsaka,isozaki}@cslab.kecl.ntt.co.jp

Abstract

This paper proposes the idea of ranking definitions of a person (a set of biographical facts) to automatically generate "Who is this?" quizzes. The definitions are ordered according to how difficult they make it to name the person. Such ranking would enable users to interactively learn about a person through dialogue with a system with improved understanding and lasting motivation, which is useful for educational systems. In our approach, we train a ranker that learns from data the appropriate ranking of definitions based on features that encode the importance of keywords in a definition as well as its content. Experimental results show that our approach is significantly better in ranking definitions than baselines that use conventional information retrieval measures such as tf*idf and pointwise mutual information (PMI).

1 Introduction

Appropriate ranking of sentences is important, as noted in sentence ordering tasks (Lapata, 2003), in effectively delivering content. Whether the task is to convey news texts or definitions, the objective is to make it easier for users to understand the content. However, just conveying it in an encyclopedia-like or temporal order may not be the best solution, considering that interaction between a system and a user improves understanding (Sugiyama et al., 1999) and that the cognitive load in receiving information is believed to correlate with memory fixation (Craik and Lockhart, 1972).

In this paper, we discuss the idea of ranking definitions as a way to present people's biographical information to users, and propose ranking definitions to automatically generate a "Who is this?" quiz. Here, we use the term 'definitions of a person' to mean a short series of biographical facts (See Fig. 1). The definitions are ordered according to how difficult they make it to name the person. The ranking also enables users to easily come up with answer candidates. The definitions are presented to users one by one as hints until users give the correct name (See Fig. 2). Although the interaction would take time, we could expect improved understanding of people's biographical information by users through their deliberation and the long lasting motivation afforded by the entertaining nature of quizzes, which is important in tutorial tasks (Baylor and Ryu, 2003).

Previous work on definition ranking has used measures such as tf*idf (Xu et al., 2004) or ranking models trained to encode the likelihood of a definition being good (Xu et al., 2005). However, such measures/models may not be suitable for quiz-style ranking. For example, a definition having a strong co-occurrence with a person may not be an easy hint when it is about a very minor detail. Certain descriptions, such as a person's birthplace, would have to come early so that users can easily start guessing who the person is. In our approach, we train a ranker that learns from data the appropriate ranking of definitions. Note that we only focus on the ranking of definitions and not on the interaction with users in this paper. We also assume that the definitions to be ranked are given.

Section 2 describes the task of ranking definitions, and Section 3 describes our approach. Section 4 describes our collection of ranking data and the ranking model training using the ranking support vector machine (SVM), and Section 5 presents the evaluation results. Section 6 summarizes and mentions future work.

2 Ranking Definitions for Quizzes

Figure 1 shows a list of definitions of Natsume Soseki, a famous Japanese novelist, in their original ranking at the encyclopedic website goo (*http://dictionary.goo.ne.jp/*) and in the quiz-style ranking we aim to achieve. Such a ranking would realize a dialogue like that in Fig. 2. At the end of the dialogue, the user would be able to associate the person and the definitions better, and it is expected that some new facts could be learned about that person.

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 117–120,
Prague, June 2007. ©2007 Association for Computational Linguistics

Original Ranking:

1. Novelist and scholar of British literature.
2. Real name: Kinnosuke.
3. Born in Ushigome, Edo.
4. Graduated from the University of Tokyo.
5. Master of early-modern literature along with Mori Ogai.
6. After the success of “I Am a Cat”, quit all teaching jobs and joined Asahi Shimbun.
7. Published masterpieces in Asahi Shimbun.
8. Familiar with Haiku, Chinese poetry, and calligraphy.
9. Works include “Botchan”, “Sanshiro”, etc.

⇓

Quiz-style Ranking:

1. Graduated from the University of Tokyo.
2. Born in Ushigome, Edo.
3. Novelist and scholar of British literature.
4. Familiar with Haiku, Chinese poetry, and calligraphy.
5. Published masterpieces in Asahi Shimbun.
6. Real name: Kinnosuke.
7. Master of early-modern literature along with Mori Ogai.
8. After the success of “I Am a Cat”, quit all teaching jobs and joined Asahi Shimbun.
9. Works include “Botchan”, “Sanshiro”, etc.

Figure 1: List of definitions of Natsume Soseki, a famous Japanese novelist, in their original ranking in the encyclopedia and in the quiz-style ranking. The definitions were translated by the authors.

Ranking definitions is closely related to definitional question answering and sentence ordering in multi-document summarization. In definitional question answering, measures related to information retrieval (IR), such as tf*idf or pointwise mutual information (PMI), have been used to rank sentences or information nuggets (Xu et al., 2004; Sun et al., 2005). Such measures are used under the assumption that outstanding/co-occurring keywords about a definiendum characterize that definiendum. However, this assumption may not be appropriate in quiz-style ranking; most content words in the definitions are already important in the IR sense, and strong co-occurrence may not guarantee high ranks for hints to be presented later because the hint can be too specific. An approach to creating a ranking model of definitions in a supervised manner using machine learning techniques has been reported (Xu et al., 2005). However, the model is only used to distinguish definitions from non-definitions on the basis of features related mainly to linguistic styles.

In multi-document summarization, the focus has been mainly on creating cohesive texts. (Lapata, 2003) uses the probability of words in adjacent sentences as constraints to maximize the coherence of all sentence-pairs in texts. Although we acknowledge that having cohesive definitions is important, since we are not creating a single text and the dialogue that we aim to achieve would involve frequent user/system interaction (Fig. 2), we do not deal with the coherence of definitions in this paper.

S1 Who is this? First hint: Graduated from the University of Tokyo.
U1 Yoshida Shigeru?
S2 No, not even close! Second hint: Born in Ushigome, Edo.
U2 I don’t know.
S3 OK. Third hint: Novelist and scholar of British literature.
U3 Murakami Haruki?
S4 Close! Fourth hint: Familiar with Haiku, Chinese poetry, and calligraphy.
U4 Mori Ogai?
S5 Very close! Fifth hint: Published masterpieces in Asahi Shimbun.
U5 Natsume Soseki?
S6 That’s right!

Figure 2: Example dialogue based on the quiz-style ranking of definitions. S stands for a system utterance and U for a user utterance.

3 Approach

Since it is difficult to know in advance what characteristics are important for quiz-style ranking, we learn the appropriate ranking of definitions from data. The approach is the same as that of (Xu et al., 2005) in that we adopt a machine learning approach for definition ranking, but is different in that what is learned is a quiz-style ranking of sentences that are already known to be good definitions.

First, we collect ranking data. For this purpose, we turn to existing encyclopedias for concise biographies. Then, we annotate the ranking. Secondly, we devise a set of features for a definition. Since the existence of keywords that have high scores in IR-related measures may suggest easy hints, we incorporate the scores of IR-related measures as features (*IR-related features*).

Certain words tend to appear before or after others in a biographical document to convey particular information about people (e.g., words describing occupations at the beginning; those describing works at the end, etc.) Therefore, we use word positions within the biography of the person in question as features (*positional features*). Biographies can be found in online resources, such as biography.com (*http://www.biography.com/*) and Wikipedia. In addition, to focus on the particular content of the definition, we use bag-of-words (BOW) features, together with semantic features (e.g., semantic categories in Nihongo Goi-Taikei (Ikehara et al., 1997) or word senses in WordNet) to complement the sparseness of BOW features. We describe the features we created in Section 4.2. Finally, we create a ranking model using a preference learning algorithm, such as the ranking SVM (Joachims, 2002), which learns ranking by reducing the pairwise ranking error.

4 Experiment

4.1 Data Collection

We collected biographies (in Japanese) from the goo encyclopedia. We first mined Wikipedia to calculate the PageRank$^{\text{TM}}$of people using the hyper-link structure. After sorting them in descending order by the PageRank score, we extracted the top-150 people for whom we could find an entry in the goo encyclopedia. Then, 11 annotators annotated rankings for each of the 150 people individually. The annotators were instructed to rank the definitions assuming that they were creating a "who is this?" quiz; i.e., to place the definition that is the most characteristic of the person in question at the end. The mean of the Kendall's coefficients of concordance for the 150 people was sufficiently high at 0.76 with a standard deviation of 0.13. Finally, taking the means of ranks given to each definition, we merged the individual rankings to create the reference rankings. An example of a reference ranking is the bottom one in Fig. 1. There are 958 definition sentences in all, with each person having approximately 6–7 definitions.

4.2 Deriving Features

We derived our IR-related features based on Mainichi newspaper articles (1991–2004) and Wikipedia articles. We used these two different sources to take into account the difference in the importance of terms depending on the text. We also used sentences, sections (for Wikipedia articles only) and documents as units to calculate document frequency, which resulted in the creation of five frequency tables: (i) Mainichi-Document, (ii) Mainichi-Sentence, (iii) Wikipedia-Document, (iv) Wikipedia-Section, and (v) Wikipedia-Sentence.

Using the five frequency tables, we calculated, for each content word (nouns, verbs, adjectives, and unknown words) in the definition, (1) frequency (the number of documents where the word is found), (2) relative frequency (frequency divided by the maximum number of documents), (3) co-occurrence frequency (the number of documents where both the word and the person's name are found), (4) relative co-occurrence frequency, and (5) PMI. Then, we took the minimum, maximum, and mean values of (1)–(5) for all content words in the definition as features, deriving 75 ($5 \times 5 \times 3$) features. Then, using the Wikipedia article (called an *entry*) for the person in question, we calculated (1)–(4) within the entry, and calculated tf*idf scores of words in the definition using the term frequency in the entry. Again, by taking the minimum, maximum, and mean values of (1)–(4) and tf*idf, we yielded 15 (5×3) features, for a total of 90 (75 + 15) IR-related features.

Positional features were derived also using the Wikipedia entry. For each word in the definition, we calculated (a) the number of times the word appears in the entry, (b) the minimum position of the word in the entry, (c) its maximum position, (d) its mean position, and (e) the standard deviation of the positions. Note that positions are either ordinal or relative; i.e., the relative position is calculated by dividing the ordinal position by the total number of words in the entry. Then, we took the minimum, maximum, and mean values of (a)–(e) for all content words in the definition as features, deriving 30 (5×2 (ordinal or relative positions) $\times 3$) features.

For the BOW features, we first parsed all our definitions with CaboCha (a Japanese morphological/dependency parser, *http://chasen.org/˜taku/software/cabocha/*) and extracted all content words to make binary features representing the existence of each content word. There are 2,156 BOW features in our data.

As for the semantic features, we used the semantic categories in Nihongo Goi-Taikei. Since there are 2,715 semantic categories, we created 2,715 features representing the existence of each semantic category in the definition. Semantic categories were assigned to words in the definition by a morphological analyzer that comes with ALT/J-E, a Japanese-English machine translation system (Ikehara et al., 1991).

In total, we have 4,991 features to represent each definition. We calculated all feature values for all definitions in our data to be used for the learning.

4.3 Training Ranking Models

Using the reference ranking data, we trained a ranking model using the ranking SVM (Joachims, 2002) (with a linear kernel) that minimizes the pairwise ranking error among the definitions of each person.

5 Evaluation

To evaluate the performance of the ranking model, following (Xu et al., 2004; Sun et al., 2005), we compared it with baselines that use only the scores of IR-related and positional features for ranking, i.e., sorting. Table 1 shows the performance of the ranking model (by the leave-one-out method, predicting the ranking of definitions of a person by other peo-

Rank	Description	Ranking Error
1	**Proposed ranking model**	**0.185**
2	Wikipedia-Sentence-PMI-max	0.299
3	Wikipedia-Section-PMI-max	0.309
4	Wikipedia-Document-PMI-max	0.312
5	Mainichi-Sentence-PMI-max	0.318
6	Mainichi-Document-PMI-max	0.325
7	Mainichi-Sentence-relative-co-occurrence-max	0.338
8	Wikipedia-Entry-ordinal-Min-max	0.338
9	Wikipedia-Sentence-relative-co-occurrence-max	0.339
10	Wikipedia-Entry-relative-Min-max	0.340
11	Wikipedia-Entry-ordinal-Mean-mean	0.342

Table 1: Performance of the proposed ranking model and that of 10 best-performing baselines.

ple's rankings) and that of the 10 best-performing baselines. The ranking error is pairwise ranking error; i.e., the rate of misordered pairs. A descriptive name is given for each baseline. For example, Wikipedia-Sentence-PMI-max means that we used the maximum PMI values of content words in the definition calculated from Wikipedia, with sentence as the unit for obtaining frequencies.

Our ranking model outperforms all of the baselines. McNemar's test showed that the difference between the proposed model and the best-performing baseline is significant ($p<0.00001$). The results also show that PMI is more effective in quiz-style ranking than any other measure. The fact that max is important probably means that the mere existence of a word that has a high PMI score is enough to raise the ranking of a hint. It is also interesting that Wikipedia gives better ranking, which is probably because people's names and related keywords are close to each other in such descriptive texts.

Analyzing the ranking model trained by the ranking SVM allows us to calculate the weights given to the features (Hirao et al., 2002). Table 2 shows the top-10 features in weights in absolute figures when all samples were used for training. It can be seen that high PMI values and words/semantic categories related to government or creation lead to easy hints, whereas semantic categories, such as birth and others (corresponding to the person in 'a person from Tokyo'), lead to early hints. This supports our intuitive notion that birthplaces should be presented early for users to start thinking about a person.

6 Summary and Future Work

This paper proposed ranking definitions of a person to automatically generate a "Who is this?" quiz. Using reference ranking data that we created manually, we trained a ranking model using a ranking SVM based on features that encode the importance of keywords in a definition as well as its content. Experimental results show that our ranking model significantly outperforms baselines that use single IR-related and positional measures for ranking. We are currently in the process of building a dialogue system that uses the quiz-style ranking for definition presentation. We are planning to examine how the different rankings affect the understanding and motivation of users.

Rank	Feature Name	Weight
1	Wikipedia-Sentence-PMI-max	0.723
2	SemCat:33 (others/someone)	-0.559
3	SemCat:186 (creator)	0.485
4	BOW:*bakufu* (feudal government)	0.451
5	SemCat:163 (sovereign/ruler/monarch)	0.422
6	Wikipedia-Document-PMI-max	0.409
7	SemCat:2391 (birth)	-0.404
8	Wikipedia-Section-PMI-max	0.402
9	SemCat:2595 (unit; e.g., numeral classifier)	0.374
10	SemCat:2606 (plural; e.g., plural form)	-0.368

Table 2: Weights of features learned for ranking definitions by the ranking SVM. SemCat denotes it is a semantic-category feature with its semantic category ID followed by the description of the category in parentheses. BOW denotes a BOW feature.

References

Amy Baylor and Jeeheon Ryu. 2003. Does the presence of image and animation enhance pedagogical agent persona? *Journal of Educational Computing Research*, 28(4):373–395.

Fergus I. M. Craik and Robert S. Lockhart. 1972. Levels of processing: A framework for memory research. *Journal of Verbal Learning and Verbal Behavior*, 11:671–684.

Tsutomu Hirao, Hideki Isozaki, Eisaku Maeda, and Yuji Matsumoto. 2002. Extracting important sentences with support vector machines. In *Proc. 19th COLING*, pages 342–348.

Satoru Ikehara, Satoshi Shirai, Akio Yokoo, and Hiromi Nakaiwa. 1991. Toward an MT system without pre-editing –Effects of new methods in ALT-J/E–. In *Proc. Third Machine Translation Summit: MT Summit III*, pages 101–106.

Satoru Ikehara, Masahiro Miyazaki, Satoshi Shirai, Akio Yokoo, Hiromi Nakaiwa, Kentaro Ogura, Yoshifumi Ooyama, and Yoshihiko Hayashi. 1997. *Goi-Taikei – A Japanese Lexicon*. Iwanami Shoten.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proc. KDD*, pages 133–142.

Mirella Lapata. 2003. Probabilistic text structuring: Experiments with sentence ordering. In *Proc. 41st ACL*, pages 545–552.

Akira Sugiyama, Kohji Dohsaka, and Takeshi Kawabata. 1999. A method for conveying the contents of written texts by spoken dialogue. In *Proc. PACLING*, pages 54–66.

Renxu Sun, Jing Jiang, Yee Fan Tan, Hang Cui, Tat-Seng Chua, and Min-Yen Kan. 2005. Using syntactic and semantic relation analysis in question answering. In *Proc. TREC*.

Jinxi Xu, Ralph Weischedel, and Ana Licuanan. 2004. Evaluation of an extraction-based approach to answering definitional questions. In *Proc. SIGIR*, pages 418–424.

Jun Xu, Yunbo Cao, Hang Li, and Min Zhao. 2005. Ranking definitions with supervised learning methods. In *Proc. WWW*, pages 811–819.

Predicting Evidence of Understanding by Monitoring User's Task Manipulation in Multimodal Conversations

Yukiko I. Nakano[†]
Yoshiko Arimoto[††]
Kazuyoshi Murata[†]
Yasuhiro Asa[†††]
Mika Enomoto[†]
Hirohiko Sagawa[†††]

[†]Tokyo University of Agriculture and Technology
2-24-16 Nakacho, Koganei-shi, Tokyo 184-8588, Japan
{nakano, kmurata, menomoto}@cc.tuat.ac.jp

[††]Tokyo University of Technology
1404-1 Katakura, Hachioji, Tokyo 192-0981, Japan
ar@mf.teu.ac.jp

[†††]Central Research Laboratory, Hitachi, Ltd.
1-280, Higashi-koigakubo Kokubunji-shi, Tokyo 185-8601, Japan
{yasuhiro.asa.mk, hirohiko.sagawa.cu}@hitachi.com

Abstract

The aim of this paper is to develop animated agents that can control multimodal instruction dialogues by monitoring user's behaviors. First, this paper reports on our Wizard-of-Oz experiments, and then, using the collected corpus, proposes a probabilistic model of fine-grained timing dependencies among multimodal communication behaviors: speech, gestures, and mouse manipulations. A preliminary evaluation revealed that our model can predict a instructor's grounding judgment and a listener's successful mouse manipulation quite accurately, suggesting that the model is useful in estimating the user's understanding, and can be applied to determining the agent's next action.

1 Introduction

In face-to-face conversation, speakers adjust their utterances in progress according to the listener's feedback expressed in multimodal manners, such as speech, facial expression, and eye-gaze. In task-manipulation situations where the listener manipulates objects by following the speaker's instructions, correct task manipulation by the listener serves as more direct evidence of understanding (Brennan 2000, Clark and Krych 2004), and affects the speaker's dialogue control strategies.

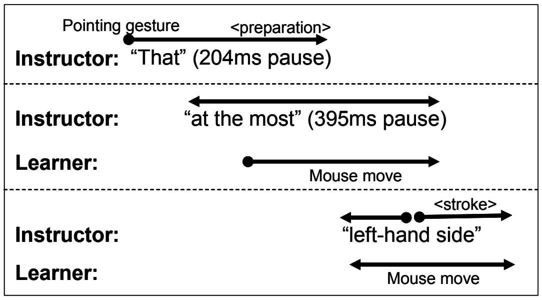


Figure 1: Example of task manipulation dialogue

Figure 1 shows an example of a software instruction dialogue in a video-mediated situation (originally in Japanese). While the learner says nothing, the instructor gives the instruction in small pieces, simultaneously modifying her gestures and utterances according to the learner's mouse movements.

To accomplish such interaction between human users and animated help agents, and to assist the user through natural conversational interaction, this paper proposes a probabilistic model that computes timing dependencies among different types of behaviors in different modalities: speech, gestures, and mouse events. The model predicts (a) whether the instructor's current utterance will be successfully understood by the learner and grounded (Clark and Schaefer 1989), and (b) whether the learner will successfully manipulate the object in the near future. These predictions can be used as constraints in determining agent actions. For example, if the current utterance will not be grounded, then the help agent must add more information.

In the following sections, first, we collect human-agent conversations by employing a Wizard-of-Oz method, and annotate verbal and nonverbal behaviors. The annotated corpus is used to build a Bayesian network model for the multimodal instruction dialogues. Finally, we will evaluate how

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 121–124,
Prague, June 2007. ©2007 Association for Computational Linguistics

accurately the model can predict the events in (a) and (b) mentioned above.

2 Related work

In their psychological study, Clark and Krych (2004) showed that speakers alter their utterances midcourse while monitoring not only the listener's vocal signals, but also the listener's gestural signals as well as through other mutually visible events. Such a bilateral process functions as a joint activity to ground the presented information, and task manipulation as a mutually visible event contributes to the grounding process (Brennan 2000, Whittaker 2003). Dillenbourg, Traum, et al. (1996) also discussed cross-modality in grounding: verbally presented information is grounded by an action in the task environment.

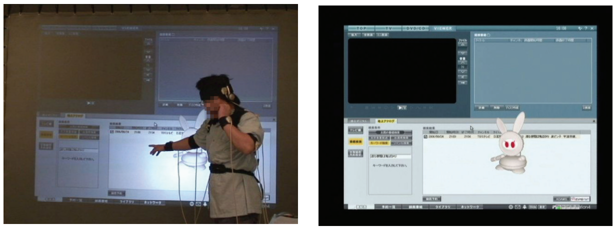
Studies on interface agents have presented computational models of multimodal interaction (Cassell, Bickmore, et al. 2000). Paek and Horvitz (1999) focused on uncertainty in speech-based interaction, and employed a Bayesian network to understand the user's speech input. For user monitoring, Nakano, Reinstein, et al. (2003) used a head tracker to build a conversational agent which can monitor the user's eye-gaze and head nods as nonverbal signals in grounding.

These previous studies provide psychological evidence about the speaker's monitoring behaviors as well as conversation modeling techniques in computational linguistics. However, little has been studied about how systems (agents) should monitor the user's task manipulation, which gives direct evidence of understanding to estimate the user's understanding, and exploits the predicted evidence as constraints in selecting the agent's next action. Based on these previous attempts, this study proposes a multimodal interaction model by focusing on task manipulation, and predicts conversation states using probabilistic reasoning.

3 Data collection

A data collection experiment was conducted using a Wizard-of-Oz agent assisting a user in learning a PCTV application, a system for watching and recording TV programs on a PC.

The output of the PC operated by the user was displayed on a 23-inch monitor in front of the user, and also projected on a 120-inch big screen, in front of which a human instructor was standing (Figure 2 (a)). Therefore, the participants shared visual events output from the PC (Figure 2 (b)) while sitting in different rooms. In addition, a rabbit-like animated agent was controlled through the instructor's motion data captured by motion sensors. The instructor's voice was changed through a voice transformation system to make it sound like a rabbit agent.



(a) Instructor (b) PC output

Figure 2: Wizard-of-Oz agent controlled by instructor

4 Corpus

We collected 20 conversations from 10 pairs, and annotated 11 conversations of 6 pairs using the Anvil video annotating tool (Kipp 2004).

Agent's verbal behaviors: The agent's (actually, instructor's) speech data was split by pauses longer than 200ms. For each inter pausal unit (IPU), utterance content type defined as follows was assigned.

- Identification (id): identification of a target object for the next operation
- Operation (op): request to execute a mouse click or a similar primitive action on the target
- Identification + operation (idop): referring to identification and operation in one IPU

In addition to these main categories, we also used: State (referring to a state before/after an operation), Function (explaining a function of the system), Goal (referring to a task goal to be accomplished), and Acknowledgment. The intercoder agreement for this coding scheme is very high K=0.89 (Cohen's Kappa), suggesting that the assigned tags are reliable.

Agent's nonverbal behaviors: As the most salient instructor's nonverbal behaviors in the collected data, we annotated agent pointing gestures:

- Agent movement: agent's position movement
- Agent touching target (att): agent's touching the target object as a stroke of a pointing gesture

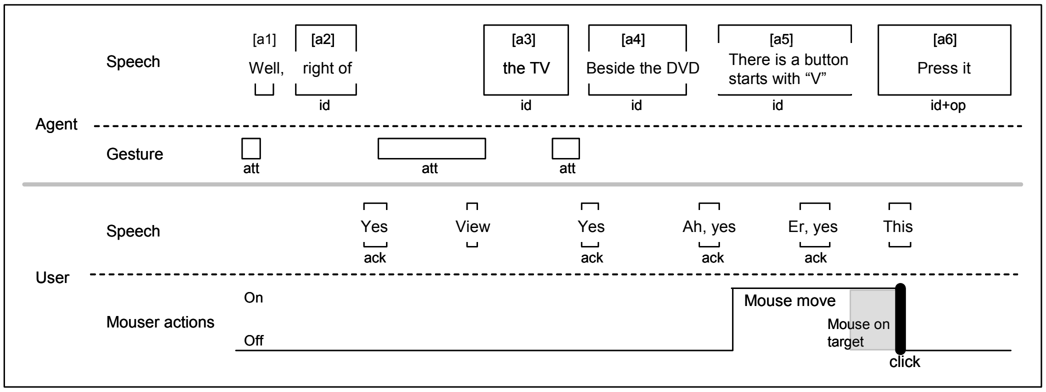


Figure 3: Example dialogue between Wizard-of-Oz agent and user

User's nonverbal behaviors: We annotated three types of mouse manipulation for the user's task manipulation as follows:

- Mouse movement: movement of the mouse cursor
- Mouse-on-target: the mouse cursor is on the target object
- Click target: click on the target object

4.1 Example of collected data

An example of an annotated corpus is shown in Figure 3. The upper two tracks illustrate the agent's verbal and nonverbal behaviors, and the other two illustrate the user's behaviors. The agent was pointing at the target (att) and giving a sequence of identification descriptions [a1-3]. Since the user's mouse did not move at all, the agent added another identification IPU [a4] accompanied by another pointing gesture. Immediately after that, the user's mouse cursor started moving towards the target object. After finishing the next IPU, the agent finally requested the user to click the object in [a6]. Note that the collected Wizard-of-Oz conversations are very similar to the human-human instruction dialogues shown in Figure 1. While carefully monitoring the user's mouse actions, the Wizard-of-Oz agent provided information in small pieces. If it was uncertain that the user was following the instruction, the agent added more explanation without continuing.

5 Probabilistic model of user-agent multimodal interaction

5.1 Building a Bayesian network model

To consider multiple factors for verbal and nonverbal behaviors in probabilistic reasoning, we employed a Bayesian network technique, which can infer the likelihood of the occurrence of a target event based on the dependencies among multiple kinds of evidence. We extracted the conversational data from the beginning of an instructor's identification utterance for a new target object to the point that the user clicks on the object. Each IPU was split at 500ms intervals, and 1395 intervals were obtained. As shown in Figure 4, the network consists of 9 properties concerning verbal and nonverbal behaviors for past, current, and future interval(s).

5.2 Predicting evidence of understanding

As a preliminary evaluation, we tested how accurately our Bayesian network model can predict an instructor's grounding judgment, and the user's mouse click. The following five kinds of evidence were given to the network to predict future states. As evidence for the previous three intervals (1.5 sec), we used (1) the percentage of time the agent touched the target (att), (2) the number of the user's mouse movements. Evidence for the current interval is (3) current IPU's content type, (4) whether the end of the current interval will be the end of the IPU (i.e. whether a pause will follow after the current interval), and (5) whether the mouse is on the target object.

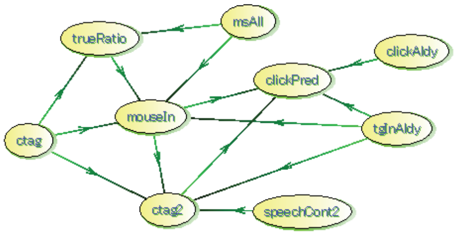


Figure 4: Bayesian network model

Table 1: Preliminary evaluation results

	Precision	Recall	F-measure
Content change	0.53	0.99	0.68
Same content	1.00	0.81	0.90

(a) Predicting grounding judgment: We tested how accurately the model can predict whether the instructor will go on to the next leg of the instruction or will give additional explanations using the same utterance content type (the current message will not be grounded).

The results of 5-fold cross-validation are shown in Table 1. Since 83% of the data are "same content" cases, prediction for "same content" is very accurate (F-measure is 0.90). However, it is not very easy to find "content change" case because of its less frequency (F-measure is 0.68). It would be better to test the model using more balanced data.

(b) Predicting user's mouse click: As a measure of the smoothness of task manipulation, the network predicted whether the user's mouse click would be successfully performed within the next 5 intervals (2.5sec). If a mouse click is predicted, the agent should just wait without annoying the user by unnecessary explanation. Since randomized data is not appropriate to test mouse click prediction, we used 299 sequences of utterances that w-ere not used for training. Our model predicted 84% of the user's mouse clicks: 80% of them were predicted 3-5 intervals before the actual occurrence of the mouse click, and 20% were predicted 1 interval before. However, the model frequently generates wrong predictions. Improving precision rate is necessary.

6 Discussion and Future Work

We employed a Bayesian network technique to our goal of developing conversational agents that can generate fine-grained multimodal instruction dialogues, and we proposed a probabilistic model for predicting grounding judgment and user's successful mouse click. The results of preliminary evaluation suggest that separate models of each modality for each conversational participant cannot properly describe the complex process of on-going multimodal interaction, but modeling the interaction as dyadic activities with multiple tracks of modalities is a promising approach.

The advantage of employing the Bayesian network technique is that, by considering the cost of misclassification and the benefit of correct classification, the model can be easily adjusted according to the purpose of the system or the user's skill level. For example, we can make the model more cautious or incautious. Thus, our next step is to implement the proposed model into a conversational agent, and evaluate our model not only in its accuracy, but also in its effectiveness by testing the model with various utility values.

References

Brennan, S. 2000. Processes that shape conversation and their implications for computational linguistics. *In Proceedings of 38th Annual Meeting of the ACL.*

Cassell, J., Bickmore, T., Campbell, L., Vilhjalmsson, H. and Yan, H. (2000). Human Conversation as a System Framework: Designing Embodied Conversational Agents. *Embodied Conversational Agents*. J. Cassell, J. Sullivan, S. Prevost and E. Churchill. Cambridge, MA, MIT Press: 29-63.

Clark, H. H. and Schaefer, E. F. 1989. Contributing to discourse. *Cognitive Science* 13: 259-294.

Clark, H. H. and Krych, M. A. 2004. Speaking while monitoring addressees for understanding. *Journal of Memory and Language* 50(1): 62-81.

Dillenbourg, P., Traum, D. R. and Schneider, D. 1996. Grounding in Multi-modal Task Oriented Collaboration. *In Proceedings of EuroAI&Education Conference*: 415-425.

Kipp, M. 2004. Gesture Generation by Imitation - From Human Behavior to Computer Character Animation, Boca Raton, Florida: Dissertation.com.

Nakano, Y. I., Reinstein, G., Stocky, T. and Cassell, J. 2003. Towards a Model of Face-to-Face Grounding. *In Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*: 553-561.

Paek, T. and Horvitz, E. (1999). Uncertainty, Utility, and Misunderstanding: A Decision-Theoretic Perspective on Grounding in Conversational Systems. *Working Papers of the AAAI Fall Symposium on Psychological Models of Communication in Collaborative Systems*. S. E. Brennan, A. Giboin and D. Traum: 85-92.

Whittaker, S. (2003). Theories and Methods in Mediated Communication. *The Handbook of Discourse Processes*. A. Graesser, MIT Press.

Automatically Assessing the Post Quality in Online Discussions on Software

Markus Weimer and **Iryna Gurevych** and **Max Mühlhäuser**
Ubiquitous Knowledge Processing Group, Division of Telecooperation
Darmstadt University of Technology, Germany
http://www.ukp.informatik.tu-darmstadt.de
[mweimer,gurevych,max]@tk.informatik.tu-darmstadt.de

Abstract

Assessing the quality of user generated content is an important problem for many web forums. While quality is currently assessed manually, we propose an algorithm to assess the quality of forum posts automatically and test it on data provided by Nabble.com. We use state-of-the-art classification techniques and experiment with five feature classes: Surface, Lexical, Syntactic, Forum specific and Similarity features. We achieve an accuracy of 89% on the task of automatically assessing post quality in the software domain using forum specific features. Without forum specific features, we achieve an accuracy of 82%.

1 Introduction

Web 2.0 leads to the proliferation of user generated content, such as blogs, wikis and forums. Key properties of user generated content are: low publication threshold and a lack of editorial control. Therefore, the quality of this content may vary. The end user has problems to navigate through large repositories of information and find information of high quality quickly. In order to address this problem, many forum hosting companies like Google Groups[1] and Nabble[2] introduce rating mechanisms, where users can rate the information manually on a scale from 1 (low quality) to 5 (high quality). The ratings have been shown to be consistent with the user community by Lampe and Resnick (2004). However, the percentage of manually rated posts is very low (0.1% in Nabble).

Departing from this, the main idea explored in the present paper is to investigate the feasibility of automatically assessing the perceived quality of user generated content. We test this idea for online forum discussions in the domain of software. The *perceived quality* is not an objective measure. Rather, it models how the community at large perceives post quality. We choose a machine learning approach to automatically assess it.

Our main contributions are: (1) An algorithm for automatic quality assessment of forum posts that learns from human ratings. We evaluate the system on online discussions in the software domain. (2) An analysis of the usefulness of different classes of features for the prediction of post quality.

2 Related work

To the best of our knowledge, this is the first work which attempts to assess the quality of forum posts automatically. However, on the one hand work has been done on automatic assessment of other types of user generated content, such as essays and product reviews. On the other hand, student online discussions have been analyzed.

Automatic text quality assessment has been studied in the area of automatic essay scoring (Valenti et al., 2003; Chodorow and Burstein, 2004; Attali and Burstein, 2006). While there exist guidelines for writing and assessing essays, this is not the case for forum posts, as different users cast their rating with possibly different quality criteria in mind. The same argument applies to the automatic assessment of product review usefulness (Kim et al., 2006c):

[1] http://groups.google.com

[2] http://www.nabble.com

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 125–128,
Prague, June 2007. ©2007 Association for Computational Linguistics

Stars	Label on the website	Number
⋆	Poor Post	1251
⋆⋆	Below Average Post	44
⋆⋆⋆	Average Post	69
⋆⋆⋆⋆	Above Average Post	183
⋆⋆⋆⋆⋆	Excellent Post	421

Table 1: Categories and their usage frequency.

Readers of a review are asked “Was this review helpful to you?” with the answer choices Yes/No. This is very well defined compared to forum posts, which are typically rated on a five star scale that does not advertise a specific semantics.

Forums have been in the focus of another track of research. Kim et al. (2006b) found that the relation between a student’s posting behavior and the grade obtained by that student can be assessed automatically. The main features used are the number of posts, the average post length and the average number of replies to posts of the student. Feng et al. (2006) and Kim et al. (2006a) describe a system to find the most authoritative answer in a forum thread. The latter add speech act analysis as a feature for this classification. Another feature is the author’s trustworthiness, which could be computed based on the automatic quality classification scheme proposed in the present paper. Finding the most authoritative post could also be defined as a special case of the quality assessment. However, it is definitely different from the task studied in the present paper. We assess the perceived quality of a given post, based solely on its intrinsic features. Any discussion thread may contain an indefinite number of good posts, rather than a single authoritative one.

3 Experiments

We seek to develop a system that adapts to the quality standards existing in a certain user community by learning the relation between a set of features and the perceived quality of posts. We experimented with features from five classes described in table 2: *Surface, Lexical, Syntactic, Forum specific and Similarity features.*

We use forum discussions from the *Software* category of Nabble.com.[5] The data consists of 1968 rated posts in 1788 threads from 497 forums. Posts can be rated by multiple users, but that happens rarely. 1927 posts were rated by one, 40 by two and 1 post by three users. Table 1 shows the distribution of average ratings on a five star scale. From this statistics, it becomes evident that users at Nabble prefer extreme ratings. Therefore, we decided to treat the posts as being binary rated.: Posts with less than three stars are rated “bad”. Posts with more than three stars are “good”.

We removed 61 posts where all ratings are exactly three stars. We removed additional 14 posts because they had contradictory ratings on the binary scale. Those posts were mostly spam, which was voted high for commercial interests and voted down for being spam. Additionally, we removed 30 posts that did not contain any text but only attachments like pictures. Finally, we removed 331 non English posts using a simple heuristics: Posts that contained a certain percentage of words above a pre-defined threshold, which are non-English according to a dictionary, were considered to be non-English.

This way, we obtained 1532 binary classified posts: 947 good posts and 585 bad posts. For each post, we compiled a feature vector, and feature values were normalized to the range $[0.0, \ldots, 1.0]$.

We use support vector machines as a state-of-the-art-algorithm for binary classification. For all experiments, we used a C-SVM with a gaussian RBF kernel as implemented by LibSVM in the YALE toolkit (Chang and Lin, 2001; Mierswa et al., 2006). Parameters were set to $C = 10$ and $\gamma = 0.1$. We performed stratified ten-fold cross validation[6] to estimate the performance of our algorithm. We repeated several experiments according to the leave-one-out evaluation scheme and found comparable results to the ones reported in this paper.

4 Results and Analysis

We compared our algorithm to a majority class classifier as a baseline, which achieves an accuracy of 62%. As it is evident from table 3, most system configurations outperform the baseline system. The best performing single feature category are the Forum specific features. As we seek to build an adaptable system, analyzing the performance without these features is worthwhile: Using all other features, we

[5] `http://www.nabble.com/Software-f94.html`

[6] See (Witten and Frank, 2005), chapter 5.3 for an in-depth description.

Feature category	Feature name	Description
Surface Features	Length	The number of tokens in a post.
	Question Frequency	The percentage of sentences ending with "?".
	Exclamation Frequency	The percentage of sentences ending with "!".
	Capital Word Frequency	The percentage of words in CAPITAL, which is often associated with shouting.
Lexical Features	Spelling Error Frequency	The percentage of words that are not spelled correctly.[3]
Information about the wording of the posts	Swear Word Frequency	The percentage of words that are on a list of swear words we compiled from resources like WordNet and Wikipedia[4], which contains more than eighty words like "asshole", but also common transcriptions like "f*ckin".
Syntactic Features		The percentage of part-of-speech tags as defined in the PENN Treebank tag set (Marcus et al., 1994). We used TreeTagger (Schmid, 1995) based on the english parameter files supplied with it.
Forum specific features	IsHTML	Whether or not a post contains HTML. In our data, this is encoded explicitly, but it can also be determined by regular expressions matching HTML tags.
Properties of a post that are only	IsMail	Whether or not a post has been copied from a mailing list. This is encoded explicitly in our data.
present in forum postings	Quote Fraction	The fraction of characters that are inside quotes of other posts. These quotes are marked explicitly in our data.
	URL and Path Count	The number of URLs and filesystem paths. Post quality in the software domain may be influenced by the amount of tangible information, which is partly captured by these features.
Similarity features		Forums are focussed on a topic. The relatedness of a post to the topic of the forum may influence post quality. We capture this relatedness by the cosine between the posts unigram vector and the unigram vector of the forum.

Table 2: Features used for the automatic quality assessment of posts.

achieve an only slightly worse classification accuracy. Thus, the combination of all other features captures the quality of a post fairly well.

SUF	LEX	SYN	FOR	SIM	Avg. accuracy
		Baseline			*61.82%*
√	√	√	√	√	89.10%
√	–	–	–	–	61.82%
–	√	–	–	–	71.82%
–	–	√	–	–	82.64%
–	–	–	√	–	85.05%
–	–	–	–	√	62.01%
–	√	√	√	√	89.10%
√	–	√	√	√	89.36%
√	√	–	√	√	85.03%
√	√	√	–	√	82.90%
√	√	√	√	–	88.97%
–	√	√	√	–	88.56%
√	–	–	√	–	85.12%
–	–	√	√	–	88.74%

Table 3: Accuracy with different feature sets. SUF: Surface, LEX: Lexical, SYN: Syntax, FOR: Forum specific, SIM: similarity. The *baseline* results from a majority class classifier.

We performed additional experiments to identify the most important features from the Forum specific ones. Table 4 shows that IsMail and Quote Fraction are the dominant features. This is noteworthy, as those features are not based on the domain of discussion. Thus, we believe that these features will perform well in future experiments on other data.

ISM	ISH	QFR	URL	PAC	Avg. accuracy
√	√	√	√	√	*85.05%*
√	–	–	–	–	73.30%
–	√	–	–	–	61.82%
–	–	√	–	–	73.76%
–	–	–	√	–	61.29%
–	–	–	–	√	61.82%
–	√	√	√	√	74.41%
√	–	√	√	√	*85.05%*
√	√	–	√	√	73.30%
√	√	√	–	√	*85.05%*
√	√	√	√	–	*85.05%*
√	–	√	–	–	84.99%
√	√	√	–	–	*85.05%*

Table 4: Accuracy with different forum specific features. ISM: IsMail, ISH: IsHTML, QFR: QuoteFraction, URL: URLCount, PAC: PathCount.

Error Analysis Table 5 shows the confusion matrix of the system using all features. Many posts that were misclassified as good ones show no apparent reason to be classified as bad posts to us. The understanding of their rating seems to require deep knowledge about the specific subject of discussion. The few remaining posts are either spam or rated negatively to signalize dissent with the opinion expressed in the post. Posts that were misclassified as bad ones often contain program code, digital signatures or other non-textual parts in the body. We plan to address these issues with better preprocessing in

	true good	true bad	sum
pred. good	490	72	562
pred. bad	95	875	970
sum	585	947	1532

Table 5: Confusion matrix for the system using all features.

the future. However, the relatively high accuracy already achieved shows that these issues are rare.

5 Conclusion and Future Work

Assessing post quality is an important problem for many forums on the web. Currently, most forums need their users to rate the posts manually, which is error prone, labour intensive and last but not least may lead to the problem of premature negative consent (Lampe and Resnick, 2004).

We proposed an algorithm that has shown to be able to assess the quality of forum posts. The algorithm applies state-of-the-art classification techniques using features such as *Surface, Lexical, Syntactic, Forum specific and Similarity features* to do so. Our best performing system configuration achieves an accuracy of 89.1%, which is significantly higher than the baseline of 61.82%. Our experiments show that forum specific features perform best. However, slightly worse but still satisfactory performance can be obtained even without those.

So far, we have not made use of the structural information in forum threads yet. We plan to perform experiments investigating speech act recognition in forums to improve the automatic quality assessment. We also plan to apply our system to further domains of forum discussion, such as the discussions among active Wikipedia users.

We believe that the proposed algorithm will support important applications beyond content filtering like automatic summarization systems and forum specific search.

Acknowledgments

This work was supported by the German Research Foundation as part of the Research Training Group "Feedback-Based Quality Management in eLearning" under the grant 1223. We are thankful to Nabble for providing their data.

References

Yigal Attali and Jill Burstein. 2006. Automated essay scoring with e-rater v.2. *The Journal of Technology, Learning, and Assessment*, 4(3), February.

Chih-Chung Chang and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Martin Chodorow and Jill Burstein. 2004. Beyond essay length: Evaluating e-raters performance on toefl essays. Technical report, ETS.

Donghui Feng, Erin Shaw, Jihie Kim, and Eduard Hovy. 2006. Learning to detect conversation focus of threaded discussions. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics (HLT-NNACL)*.

Jihie Kim, Grace Chern, Donghui Feng, Erin Shaw, and Eduard Hovya. 2006a. Mining and assessing discussions on the web through speech act analysis. In *Proceedings of the Workshop on Web Content Mining with Human Language Technologies at the 5th International Semantic Web Conference*.

Jihie Kim, Erin Shaw, Donghui Feng, Carole Beal, and Eduard Hovy. 2006b. Modeling and assessing student activities in on-line discussions. In *Proceedings of the Workshop on Educational Data Mining at the conference of the American Association of Artificial Intelligence (AAAI-06)*, Boston, MA.

Soo-Min Kim, Patrick Pantel, Tim Chklovski, and Marco Pennacchiotti. 2006c. Automatically assessing review helpfulness. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 423 – 430, Sydney, Australia, July.

Cliff Lampe and Paul Resnick. 2004. Slash(dot) and burn: Distributed moderation in a large online conversation space. In *Proceedings of ACM CHI 2004 Conference on Human Factors in Computing Systems, Vienna Austria*, pages 543–550.

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1994. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330.

Ingo Mierswa, Michael Wurst, Ralf Klinkenberg, Martin Scholz, and Timm Euler. 2006. YALE: Rapid prototyping for complex data mining tasks. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 935–940, New York, NY, USA. ACM Press.

Helmut Schmid. 1995. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *International Conference on New Methods in Language Processing*, Manchester, UK.

Salvatore Valenti, Francesca Neri, and Alessandro Cucchiarelli. 2003. An overview of current research on automated essay grading. *Journal of Information Technology Education*, 2:319–329.

Ian H. Witten and Eibe Frank. 2005. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, San Francisco, 2 edition.

WordNet-based Semantic Relatedness Measures in Automatic Speech Recognition for Meetings

Michael Pucher
Telecommunications Research Center Vienna
Vienna, Austria
Speech and Signal Processing Lab, TU Graz
Graz, Austria
pucher@ftw.at

Abstract

This paper presents the application of WordNet-based semantic relatedness measures to *Automatic Speech Recognition* (ASR) in multi-party meetings. Different word-utterance context relatedness measures and utterance-coherence measures are defined and applied to the rescoring of N-best lists. No significant improvements in terms of *Word-Error-Rate* (WER) are achieved compared to a large word-based n-gram baseline model. We discuss our results and the relation to other work that achieved an improvement with such models for simpler tasks.

1 Introduction

As (Pucher, 2005) has shown different WordNet-based measures and contexts are best for word prediction in conversational speech. The JCN (Section 2.1) measure performs best for nouns using the noun-context. The LESK (Section 2.1) measure performs best for verbs and adjectives using a mixed word-context.

Text-based semantic relatedness measures can improve word prediction on simulated speech recognition hypotheses as (Demetriou et al., 2000) have shown. (Demetriou et al., 2000) generated N-best lists from phoneme confusion data acquired from a speech recognizer, and a pronunciation lexicon. Then sentence hypotheses of varying *Word-Error-Rate* (WER) were generated based on sentences from different genres from the *British National Corpus* (BNC). It was shown by them that the semantic model can improve recognition, where the amount of improvement varies with context length and sentence length. Thereby it was shown that these models can make use of long-term information.

In this paper the best performing measures from (Pucher, 2005), which outperform baseline models on word prediction for conversational telephone speech are used for *Automatic Speech Recognition* (ASR) in multi-party meetings. Thereby we want to investigate if WordNet-based models can be used for rescoring of ‘real’ N-best lists in a difficult task.

1.1 Word prediction by semantic similarity

The standard n-gram approach in language modeling for speech recognition cannot cope with long-term dependencies. Therefore (Bellegarda, 2000) proposed combining n-gram language models, which are effective for predicting local dependencies, with *Latent Semantic Analysis* (LSA) based models for covering long-term dependencies. WordNet-based semantic relatedness measures can be used for word prediction using long-term dependencies, as in this example from the CallHome English telephone speech corpus:

(1) B: I I well, you should see what the ⌊students⌋
B: after they torture them for six ⌊years⌋ in middle ⌊school⌋ and high ⌊school⌋ they don’t want to do anything in ⌊college⌋ particular.

In Example 1 *college* can be predicted from the noun context using semantic relatedness measures,

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 129–132,
Prague, June 2007.

here between *students* and *college*. A 3-gram model gives a ranking of *college* in the context of *anything in*. An 8-gram predicts *college* from *they don't want to do anything in*, but the strongest predictor is *students*.

1.2 Test data

The JCN and LESK measure that are defined in the next section are used for N-best list rescoring. For the WER experiments N-best lists generated from the decoding of conference room meeting test data of the NIST Rich Transcription 2005 Spring (RT-05S) meeting evaluation (Fiscus et al., 2005) are used. The 4-gram that has to be improved by the WordNet-based models is trained on various corpora from conversational telephone speech to web data that together contain approximately 1 billion words.

2 WordNet-based semantic relatedness measures

2.1 Basic measures

Two similarity/distance measures from the Perl package WordNet-Similarity written by (Pedersen et al., 2004) are used. The measures are named after their respective authors. All measures are implemented as similarity measures. JCN (Jiang and Conrath, 1997) is based on the information content, and LESK (Banerjee and Pedersen, 2003) allows for comparison across Part-of-Speech (POS) boundaries.

2.2 Word context relatedness

First the relatedness between words is defined based on the relatedness between senses. $\mathrm{S}(w)$ are the senses of word w. Definition 2 also performs word-sense disambiguation.

$$\mathrm{rel}(w, w') = \max_{c_i \in \mathrm{S}(w)\ c_j \in \mathrm{S}(w')} \mathrm{rel}(c_i, c_j) \qquad (2)$$

The relatedness of a word and a context (rel_W) is defined as the average of the relatedness of the word and all words in the context.

$$\mathrm{rel}_\mathrm{W}(w, C) = \frac{1}{\mid C \mid} \sum_{w_i \in C} \mathrm{rel}(w, w_i) \qquad (3)$$

2.3 Word utterance (context) relatedness

The performance of the word-context relatedness (Definition 3) shows how well the measures work for algorithms that proceed in a left-to-right manner, since the context is restricted to words that have already been seen. For the rescoring of N-best lists it is not necessary to proceed in a left-to-right manner. The word-utterance-context relatedness can be used for the rescoring of N-best lists. This relatedness does not only use the context of the preceding words, but the whole utterance.

Suppose $U = \langle w_1, \ldots, w_n \rangle$ is an utterance. Let $\mathrm{pre}(w_i, U)$ be the set $\bigcup_{j<i} w_j$ and $\mathrm{post}(w_i, U)$ be the set $\bigcup_{j>i} w_j$. Then the word-utterance-context relatedness is defined as

$$\begin{aligned}&\mathrm{rel}_{\mathrm{U}_1}(w_i, U, C) = \\ &\quad \mathrm{rel}_\mathrm{W}(w_i, \mathrm{pre}(w_i, U) \cup \mathrm{post}(w_i, U) \cup C)\ . \end{aligned} \qquad (4)$$

In this case there are two types of context. The first context comes from the respective meeting, and the second context comes from the actual utterance.

Another definition is obtained if the context C is eliminated ($C = \emptyset$) and just the utterance context U is taken into account.

$$\begin{aligned}&\mathrm{rel}_{\mathrm{U}_2}(w_i, U) = \\ &\quad \mathrm{rel}_\mathrm{W}(w_i, \mathrm{pre}(w_i, U) \cup \mathrm{post}(w_i, U)) \end{aligned} \qquad (5)$$

Both definitions can be modified for usage with rescoring in a left-to-right manner by restricting the contexts only to the preceding words.

$$\mathrm{rel}_{\mathrm{U}_3}(w_i, U, C) = \mathrm{rel}_\mathrm{W}(w_i, \mathrm{pre}(w_i, U) \cup C) \qquad (6)$$

$$\mathrm{rel}_{\mathrm{U}_4}(w_i, U) = \mathrm{rel}_\mathrm{W}(w_i, \mathrm{pre}(w_i, U)) \qquad (7)$$

2.4 Defining utterance coherence

Using Definitions 4-7 different concepts of utterance coherence can be defined. For rescoring the utterance coherence is used, when a score for each element of an N-best list is needed. U is again an utterance $U = \langle w_1, \ldots, w_n \rangle$.

$$\mathrm{cohU}_1(U,C) = \frac{1}{\mid U \mid} \sum_{w \in U} \mathrm{rel}_{\mathrm{U}_1}(w,U,C) \quad (8)$$

The first semantic utterance coherence measure (Definition 8) is based on all words in the utterance as well as in the context. It takes the mean of the relatedness of all words. It is based on the word-utterance-context relatedness (Definition 4).

$$\mathrm{cohU}_2(U) = \frac{1}{\mid U \mid} \sum_{w \in U} \mathrm{rel}_{\mathrm{U}_2}(w,U) \quad (9)$$

The second coherence measure (Definition 9) is a pure inner-utterance-coherence, which means that no history apart from the utterance is needed. Such a measure is very useful for rescoring, since the history is often not known or because there are speech recognition errors in the history. It is based on Definition 5.

$$\mathrm{cohU}_3(U,C) = \frac{1}{\mid U \mid} \sum_{w \in U} \mathrm{rel}_{\mathrm{U}_3}(w,U,C) \quad (10)$$

The third (Definition 10) and fourth (Definition 11) definition are based on Definition 6 and 7, that do not take future words into account.

$$\mathrm{cohU}_4(U) = \frac{1}{\mid U \mid} \sum_{w \in U} \mathrm{rel}_{\mathrm{U}_4}(w,U) \quad (11)$$

3 Word-error-rate (WER) experiments

For the rescoring experiments the first-best element of the previous N-best list is added to the context. Before applying the WordNet-based measures, the N-best lists are POS tagged with a decision tree tagger (Schmid, 1994). The WordNet measures are then applied to verbs, nouns and adjectives. Then the similarity values are used as scores, which have to be combined with the language model scores of the N-best list elements.

The JCN measure is used for computing a noun score based on the noun context, and the LESK measure is used for computing a verb/adjective score based on the noun/verb/adjective context. In the end there is a *lesk_score* and a *jcn_score* for each N-best list. The final WordNet score is the sum of the two scores.

The log-linear interpolation method used for the rescoring is defined as

$$p(S) \propto p_{\mathrm{wordnet}}(S)^{\lambda}\, p_{n\text{-gram}}(S)^{1-\lambda} \quad (12)$$

where $\propto$ denotes normalization. Based on all WordNet scores of an N-best list a probability is estimated, which is then interpolated with the n-gram model probability. If only the elements in an N-best list are considered, log-linear interpolation can be used since it is not necessary to normalize over all sentences. Then there is only one parameter λ to optimize, which is done with a brute force approach. For this optimization a small part of the test data is taken and the WER is computed for different values of λ.

As a baseline the n-gram mixture model trained on all available training data (≈ 1 billion words) is used. It is log-linearly interpolated with the WordNet probabilities. Additionally to this sophisticated interpolation, solely the WordNet scores are used without the n-gram scores.

3.1 WER experiments for inner-utterance coherence

In this first group of experiments Definitions 8 and 9 are applied to the rescoring task. Similarity scores for each element in an N-best list are derived according to the definitions. The first-best element of the last list is always added to the context. The context size is constrained to the last 20 words. Definition 8 includes context apart from the utterance context, Definition 9 only uses the utterance context.

No improvement over the n-gram baseline is achieved for these two measures. Neither with the log-linearly interpolated models nor with the WordNet scores alone. The differences between the methods in terms of WER are not significant.

3.2 WER experiments for utterance coherence

In the second group of experiments Definitions 10 and 11 are applied to the rescoring task. There is again one measure that uses dialog context (10) and one that only uses utterance context (11).

Also for these experiments no improvement over the n-gram baseline is achieved. Neither with the log-linearly interpolated models nor with the WordNet scores alone. The differences between the methods in terms of WER are also not significant. There are also no significant differences in performance between the second group and the first group of experiments.

4 Summary and discussion

We showed how to define more and more complex relatedness measures on top of the basic relatedness measures between word senses.

The LESK and JCN measures were used for the rescoring of N-best lists. It was shown that speech recognition of multi-party meetings cannot be improved compared to a 4-gram baseline model, when using WordNet models.

One reason for the poor performance of the models could be that the task of rescoring simulated N-best lists, as presented in (Demetriou et al., 2000), is significantly easier than the rescoring of 'real' N-best lists. (Pucher, 2005) has shown that WordNet models can outperform simple random models on the task of word prediction, in spite of the noise that is introduced through word-sense disambiguation and POS tagging. To improve the word-sense disambiguation one could use the approach proposed by (Basili et al., 2004).

In the above WER experiments a 4-gram baseline model was used, which was trained on nearly 1 billion words. In (Demetriou et al., 2000) a simpler baseline has been used. 650 sentences were used there to generate sentence hypotheses with different WER using phoneme confusion data and a pronunciation lexicon. Experiments with simpler baseline models ignore that these simpler models are not used in today's recognition systems.

We think that these prediction models can still be useful for other tasks where only small amounts of training data are available. Another possibility of improvement is to use other interpolation techniques like the maximum entropy framework. WordNet-based models could also be improved by using a trigger-based approach. This could be done by not using the whole WordNet and its similarities, but defining word-trigger pairs that are used for rescoring.

5 Acknowledgements

This work was supported by the European Union 6th FP IST Integrated Project AMI (Augmented Multi-party Interaction, and by Kapsch Carrier-Com AG and Mobilkom Austria AG together with the Austrian competence centre programme **K***plus*.

References

Satanjeev Banerjee and Ted Pedersen. 2003. Extended gloss overlaps as a measure of semantic relatedness. In *Proceedings of the 18th Int. Joint Conf. on Artificial Intelligence*, pages 805–810, Acapulco.

Roberto Basili, Marco Cammisa, and Fabio Massimo Zanzotto. 2004. A semantic similarity measure for unsupervised semantic tagging. In *Proc. of the Fourth International Conference on Language Resources and Evaluation (LREC2004)*, Lisbon, Portugal.

Jerome Bellegarda. 2000. Large vocabulary speech recognition with multispan statistical language models. *IEEE Transactions on Speech and Audio Processing*, 8(1), January.

G. Demetriou, E. Atwell, and C. Souter. 2000. Using lexical semantic knowledge from machine readable dictionaries for domain independent language modelling. In *Proc. of LREC 2000, 2nd International Conference on Language Resources and Evaluation*.

Jonathan G. Fiscus, Nicolas Radde, John S. Garofolo, Audrey Le, Jerome Ajot, and Christophe Laprun. 2005. The rich transcription 2005 spring meeting recognition evaluation. In *Rich Transcription 2005 Spring Meeting Recognition Evaluation Workshop*, Edinburgh, UK.

Jay J. Jiang and David W. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the International Conference on Research in Computational Linguistics*, Taiwan.

Ted Pedersen, S. Patwardhan, and J. Michelizzi. 2004. WordNet::Similarity - Measuring the relatedness of concepts. In *Proc. of Fifth Annual Meeting of the North American Chapter of the ACL (NAACL-04)*, Boston, MA.

Michael Pucher. 2005. Performance evaluation of WordNet-based semantic relatedness measures for word prediction in conversational speech. In *IWCS 6, Sixth International Workshop on Computational Semantics*, Tilburg, Netherlands.

H Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, Manchester, UK, September.

Building Emotion Lexicon from Weblog Corpora

Changhua Yang **Kevin Hsin-Yih Lin** **Hsin-Hsi Chen**
Department of Computer Science and Information Engineering
National Taiwan University
#1 Roosevelt Rd. Sec. 4, Taipei, Taiwan 106
{d91013, f93141, hhchen}@csie.ntu.edu.tw

Abstract

An emotion lexicon is an indispensable resource for emotion analysis. This paper aims to mine the relationships between words and emotions using weblog corpora. A collocation model is proposed to learn emotion lexicons from weblog articles. Emotion classification at sentence level is experimented by using the mined lexicons to demonstrate their usefulness.

1 Introduction

Weblog (blog) is one of the most widely used cybermedia in our internet lives that captures and shares moments of our day-to-day experiences, anytime and anywhere. Blogs are web sites that timestamp posts from an individual or a group of people, called bloggers. Bloggers may not follow formal writing styles to express emotional states. In some cases, they must post in pure text, so they add printable characters, such as ":-)" (happy) and ":-(" (sad), to express their feelings. In other cases, they type sentences with an internet messenger-style interface, where they can attach a special set of graphic icons, or emoticons. Different kinds of emoticons are introduced into text expressions to convey bloggers' emotions.

Since thousands of blog articles are created everyday, emotional expressions can be collected to form a large-scale corpus which guides us to build vocabularies that are more emotionally expressive. Our approach can create an emotion lexicon free of laborious efforts of the experts who must be familiar with both linguistic and psychological knowledge.

2 Related Works

Some previous works considered emoticons from weblogs as categories for text classification. Mishne (2005), and Yang and Chen (2006) used emoticons as tags to train SVM (Cortes and Vapnik, 1995) classifiers at document or sentence level. In their studies, emoticons were taken as moods or emotion tags, and textual keywords were taken as features. Wu et al. (2006) proposed a sentence-level emotion recognition method using dialogs as their corpus. "Happy, "Unhappy", or "Neutral" was assigned to each sentence as its emotion category. Yang et al. (2006) adopted Thayer's model (1989) to classify music emotions. Each music segment can be classified into four classes of moods. In sentiment analysis research, Read (2005) used emoticons in newsgroup articles to extract instances relevant for training polarity classifiers.

3 Training and Testing Blog Corpora

We select Yahoo! Kimo Blog[1] posts as our source of emotional expressions. Yahoo! Kimo Blog service has 40 emoticons which are shown in Table 1. When an editing article, a blogger can insert an emoticon by either choosing it or typing in the corresponding codes. However, not all articles contain emoticons. That is, users can decide whether to insert emoticons into articles/sentences or not. In this paper, we treat these icons as emotion categories and taggings on the corresponding text expressions.

The dataset we adopt consists of 5,422,420 blog articles published at Yahoo! Kimo Blog from January to July, 2006, spanning a period of 212 days. In total, 336,161 bloggers' articles were collected. Each blogger posts 16 articles on average.

We used the articles from January to June as the training set and the articles in July as the testing set. Table 2 shows the statistics of each set. On average, 14.10% of the articles contain emotion-tagged expressions. The average length of articles with tagged emotions, i.e., 272.58 characters, is shorter

[1] http://tw.blog.yahoo.com/

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 133–136,
Prague, June 2007. ©2007 Association for Computational Linguistics

Table 1. Yahoo! Kimo Blog Emoticon Set.

ID	Emoticon	Code	Description	ID	Emoticon	Code	Description	ID	Emoticon	Code	Description	ID	Emoticon	Code	Description
1		:)	happy	11		:O	surprise	21		0:)	angel	31		(:|	yawn
2		:(	sad	12		X-(	angry	22		:-B	nerd	32		=P~	drooling
3		;)	winking	13		:>	smug	23		=;	talk to the hand	33		:-?	thinking
4		:D	big grin	14		B-)	cool	24		I-)	asleep	34		;))	hee hee
5		;;)	batting eyelashes	15		:-S	worried	25		8-)	rolling eyes	35		=D>	applause
6		:-/	confused	16		>:)	devil	26		:-&	sick	36		[-o<	praying
7		:x	love struck	17		:((	crying	27		:-$	don't tell anyone	37		:-<	sigh
8		:”>	blushing	18		:))	laughing	28		[-(	not talking	38		>:P	phbbbbt
9		:p	tongue	19		:|	straight face	29		:o)	clown	39		@};-	rose
10		:*	kiss	20		/:)	raised eyebrow	30		@-)	hypnotized	40		:@)	pig

Table 2. Statistics of the Weblog Dataset.

Dataset	Article #	Tagged #	Percentage	Tagged Len.	Untagged L.
Training	4,187,737	575,009	13.86%	269.77 chrs.	468.14 chrs.
Testing	1,234,683	182,999	14.92%	281.42 chrs.	455.82 chrs.
Total	5,422,420	764,788	14.10%	272.58 chrs.	465.37 chrs.

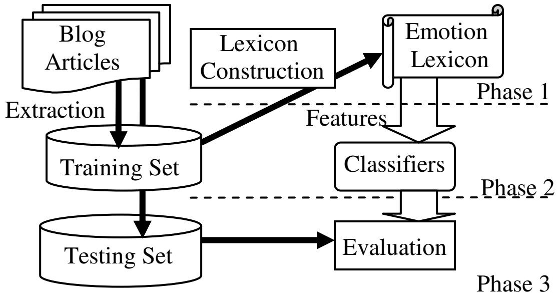


Figure 1. Emotion Lexicon Construction and Evaluation.

than that of articles without tagging, i.e., 465.37 characters. It seems that people tend to use emoticons to replace certain amount of text expressions to make their articles more succinct.

Figure 1 shows the three phases for the construction and evaluation of emotion lexicons. In phase 1, 1,185,131 sentences containing only one emoticon are extracted to form a training set to build emotion lexicons. In phase 2, sentence-level emotion classifiers are constructed using the mined lexicons. In phase 3, a testing set consisting of 307,751 sentences is used to evaluate the classifiers.

4 Emotion Lexicon Construction

The blog corpus contains a collection of bloggers' emotional expressions which can be analyzed to construct an emotion lexicon consisting of words that collocate with emoticons. We adopt a variation of pointwise mutual information (Manning and Schütze, 1999) to measure the collocation strength *co*(*e*,*w*) between an emotion *e* and a word *w*:

$$\mathrm{co}(e,w) = c(e,w) \times \log \frac{P(e,w)}{P(e)P(w)} \qquad (1)$$

where $P(e,w)=c(e,w)/N$, $P(e)=c(e)/N$, $P(w)=c(w)/N$, $c(e)$ and $c(w)$ are the total occurrences of emoticon e and word w in a tagged corpus, respectively, $c(e,w)$ is total co-occurrences of e and w, and N denotes the total word occurrences.

A word entry of a lexicon may contain several emotion senses. They are ordered by the collocation strength *co*. Figure 2 shows two Chinese example words, “哈哈” (ha1ha1) and “可惡” (ke3wu4). The former collocates with “laughing” and “big grin” emoticons with collocation strength 25154.50 and 2667.11, respectively. Similarly, the latter collocates with “angry” and “phbbbbt”. When all collocations (i.e., word-emotion pairs) are listed in a descending order of *co*, we can choose top *n* collocations to build an emotion lexicon. In this paper, two lexicons (Lexicons A and B) are extracted by setting *n* to 25k and 50k. Lexicon A contains 4,776 entries with 25,000 sense pairs and Lexicon B contains 11,243 entries and 50,000 sense pairs.

5 Emotion Classification

Suppose a sentence *S* to be classified consists of *n* emotion words. The emotion of *S* is derived by a mapping from a set of *n* emotion words to *m* emotion categories as follows:

$$S \rightarrow \{ew_1,\ldots,ew_n\} \underset{classification}{\longrightarrow} \hat{e} \in \{e_1,\ldots,e_m\}$$

哈哈 (ha1ha1) "*hah hah*"

Sense 1. 😆(laughing) – ***co***: 25154.50

e.g., 哈哈...😆 我應該要出運了~

"*hah hah...* 😆 *I am getting lucky~*"

Sense 2. 😁(big grin) – ***co***: 2667.11

e.g., 今天只背了單母音而已~哈哈 😁

"*I only memorized vowels today~ haha*😁"

可惡 (ke3wu4) "*darn*"

Sense 1. 😠(angry) – ***co***: 2797.82

e.g., 駭客在搞什麼...可惡😠

"*What's the hacker doing... darn it*😠"

Sense 2. 😒(phbbbbt) – ***co***: 619.24

e.g., 可惡的外星人...😒

"*Damn those aliens*😒"

Figure 2. Some Example Words in a Lexicon.

For each emotion word ew_i, we may find several emotion senses with the corresponding collocation strength *co* by looking up the lexicon. Three alternatives are proposed as follows to label a sentence S with an emotion:

(a) Method 1

(1) Consider all senses of ew_i as votes. Label *S* with the emotion that receives the most votes.

(2) If more than two emotions get the same number of votes, then label *S* with the emotion that has the maximum *co*.

(b) Method 2

Collect emotion senses from all ew_i. Label S with the emotion that has the maximum *co*.

(c) Method 3

The same as Method 1 except that each ew_i votes only one sense that has the maximum *co*.

In past research, the approach used by Yang et al. (2006) was based on the Thayer's model (1989), which divided emotions into 4 categories. In sentiment analysis research, such as Read's study (2006), a polarity classifier separated instances into positive and negative classes. In our experiments, we not only adopt fine-grain classification, but also coarse-grain classification. We first select 40 emoticons as a category set, and also adopt the Thayer's model to divide the emoticons into 4 quadrants of the emotion space. As shown in Figure 3, the top-right side collects the emotions that are more positive and energetic and the bottom-left side is more negative and silent. A polarity classifier uses the right side as positive and the left side as negative.

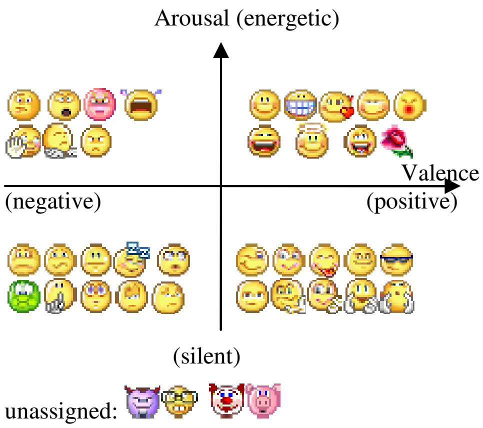


Figure 3. Emoticons on Thayer's model.

6 Evaluation

Table 3 shows the performance under various combinations of lexicons, emotion categories and classification methods. "Hit #" stands for the number of correctly-answered instances. The baseline represents the precision of predicting the majority category, such as "happy" or "positive", as the answer. The baseline method's precision increases as the number of emotion classes decreases. The upper bound recall indicates the upper limit on the fraction of the 307,751 instances solvable by the corresponding method and thus reflects the limitation of the method. The closer a method's actual recall is to the upper bound recall, the better the method. For example, at most 40,855 instances (14.90%) can be answered using Method 1 in combination with Lexicon A. But the actual recall is 4.55% only, meaning that Method 1's recall is more than 10% behind its upper bound. Methods which have a larger set of candidate answers have higher upper bound recalls, because the probability that the correct answer is in their set of candidate answers is greater.

Experiment results show that all methods utilizing Lexicon A have performance figures lower than the baseline, so Lexicon A is not useful. In contrast, Lexicon B, which provides a larger collection of vocabularies and emotion senses, outperforms Lexicon A and the baseline. Although Method 3 has the smallest candidate answer set and thus has the smallest upper bound recall, it outperforms the other two methods in most cases. Method 2 achieves better precisions when using

Table 3. Evaluation Results.

	Baseline	Method 1 (M1)				Method 2 (M2)				Method 3 (M3)			
		Upp. R.	Hit #	Prec.	Reca.	Upp. R.	Hit #	Prec.	Reca.	Upp. R.	Hit #	Prec.	Reca.
Lexicon A 40 classes	**8.04%**	14.90%	14,009	4.86%	4.55%	14.90%	9,392	3.26%	3.05%	6.49%	13,929	4.83%	4.52%
Lexicon A Thayer	**38.38%**	48.70%	90,332	32.46%	29.35%	48.70%	64,689	23.25%	21.02%	35.94%	93,285	33.53%	30.31%
Lexicon A Polarity	**63.49%**	60.74%	150,946	54.25%	49.05%	60.74%	120,237	43.21%	39.07%	54.97%	153,292	55.09%	49.81%
Lexicon B 40 classes	8.04%	73.18%	45,075	15.65%	14.65%	73.18%	43,637	15.15%	14.18%	27.89%	45,604	**15.83%**	14.81%
Lexicon B Thayer	38.38%	89.11%	104,094	37.40%	33.82%	89.11%	118,392	**42.55%**	38.47%	63.74%	110,904	39.86%	36.04%
Lexicon B Polarity	63.49%	91.12%	192,653	69.24%	62.60%	91.12%	188,434	67.72%	61.23%	81.92%	195,190	**70.15%**	63.42%

Upp. R. – upper bound recall; **Prec.** – precision; **Reca.** – recall

Table 4. SVM Performance.

Method	Upp. R.	Hit #	Prec.	Reca.	F
Lexicon B M3	81.92%	195,190	70.15%	63.42%	66.62%
SVM 25 features	15.80%	38,651	79.49%	12.56%	21.69%
SVM 50 features	26.27%	62,999	77.93%	20.47%	32.42%
SVM 75 features	36.74%	84,638	74.86%	27.50%	40.23%
SVM 100 features	45.49%	101,934	72.81%	33.12%	45.53%
(Svm-25 + M3)	90.41%	196,147	70.05%	63.73%	66.74%
(Svm-50 + M3)	90.41%	195,835	70.37%	63.64%	66.83%
(Svm-75 + M3)	90.41%	195,229	70.16%	63.44%	66.63%
(Svm-100 + M3)	90.41%	195,054	70.01%	63.38%	66.53%

F = 2×(Precision×Recall)/(Precision+Recall)

Thayer's emotion categories. Method 1 treats the vote to every sense equally. Hence, it loses some differentiation abilities. Method 1 performs the best in the first case (Lexicon A, 40 classes).

We can also apply machine learning to the dataset to train a high-precision classification model. To experiment with this idea, we adopt LIBSVM (Fan et al., 2005) as the SVM kernel to deal with the binary polarity classification problem. The SVM classifier chooses top k (k = 25, 50, 75, and 100) emotion words as features. Since the SVM classifier uses a small feature set, there are testing instances which do not contain any features seen previously by the SVM classifier. To deal with this problem, we use the class prediction from Method 3 for any testing instances without any features that the SVM classifier can recognize. In Table 4, the SVM classifier employing 25 features has the highest precision. On the other hand, the SVM classifier employing 50 features has the highest F measure when used in conjunction with Method 3.

7 Conclusion and Future Work

Our methods for building an emotional lexicon utilize emoticons from blog articles collaboratively contributed by bloggers. Since thousands of blog articles are created everyday, we expect the set of emotional expressions to keep expanding. In the experiments, the method of employing each emotion word to vote only one emotion category achieves the best performance in both fine-grain and coarse-grain classification.

Acknowledgment

Research of this paper was partially supported by Excellent Research Projects of National Taiwan University, under the contract of 95R0062-AE00-02. We thank Yahoo! Taiwan Inc. for providing the dataset for researches.

References

Corinna Cortes and V. Vapnik. 1995. Support-Vector Network. *Machine Learning*, 20:273–297.

Rong-En Fan, Pai-Hsuen Chen and Chih-Jen Lin. 2005. Working Set Selection Using Second Order Information for Training Support Vector Machines. *Journal of Machine Learning Research*, 6:1889–1918.

Gilad Mishne. 2005. Experiments with Mood Classification in Blog Posts. *Proceedings of 1st Workshop on Stylistic Analysis of Text for Information Access.*

Jonathon Read. 2005. Using Emotions to Reduce Dependency in Machine Learning Techniques for Sentiment Classification. *Proceedings of the ACL Student Research Workshop*, 43-48.

Robert E. Thayer. 1989. *The Biopsychology of Mood and Arousal, Oxford University Press.*

Changhua Yang and Hsin-Hsi Chen. 2006. A Study of Emotion Classification Using Blog Articles. *Proceedings of Conference on Computational Linguistics and Speech Processing*, 253-269.

Yi-Hsuan Yang, Chia-Chu Liu, and Homer H. Chen. 2006. Music Emotion Classification: A Fuzzy Approach. *Proceedings of ACM Multimedia*, 81-84.

Chung-Hsien Wu, Ze-Jing Chuang, and Yu-Chung Lin. 2006. Emotion Recognition from Text Using Semantic Labels and Separable Mixture Models. *ACM Transactions on Asian Language Information Processing*, 5(2):165-182.

Construction of Domain Dictionary for Fundamental Vocabulary

Chikara Hashimoto
Faculty of Engineering,
Yamagata University
4-3-16 Jonan, Yonezawa-shi, Yamagata,
992-8510 Japan

Sadao Kurohashi
Graduate School of Informatics,
Kyoto University
36-1 Yoshida-Honmachi, Sakyo-ku, Kyoto,
606-8501 Japan

Abstract

For natural language understanding, it is essential to reveal semantic relations between words. To date, only the IS-A relation has been publicly available. Toward deeper natural language understanding, we semi-automatically constructed the domain dictionary that represents the domain relation between Japanese fundamental words. This is the first Japanese domain resource that is fully available. Besides, our method does not require a document collection, which is indispensable for keyword extraction techniques but is hard to obtain. As a task-based evaluation, we performed blog categorization. Also, we developed a technique for estimating the domain of unknown words.

1 Introduction

We constructed a lexical resource that represents the domain relation among Japanese fundamental words (JFWs), and we call it the **domain dictionary**.[1] It associates JFWs with domains in which they are typically used. For example, ホームラン *home run* is associated with the domain SPORTS[2]. That is, we aim to make explicit the horizontal relation between words, the domain relation, while thesauri indicate the vertical relation called IS-A.[3]

2 Two Issues

You have to address two issues. One is what domains to assume, and the other is how to associate words with domains without document collections.

The former is paraphrased as how people categorize the real world, which is really a hard problem. In this study, we avoid being too involved in the problem and adopt a simple domain system that most people can agree on, which is as follows:

CULTURE
RECREATION
SPORTS
HEALTH
LIVING
DIET
TRANSPORTATION
EDUCATION
SCIENCE
BUSINESS
MEDIA
GOVERNMENT

It has been created based on web directories such as Open Directory Project with some adjustments. In addition, NODOMAIN was prepared for those words that do not belong to any particular domain.

As for the latter issue, you might use keyword extraction techniques; identifying words that represent a domain from the document collection using statistical measures like TF*IDF and matching between extracted words and JFWs. However, you will find that document collections of common domains such as those assumed here are hard to obtain.[4] Hence, we had to develop a method that does not require document collections. The next section details it.

[1] In fact, there have been a few domain resources in Japanese like Yoshimoto et al. (1997). But they are not publicly available.

[2] Domains are CAPITALIZED in this paper.

[3] The lack of the horizontal relationship is also known as the "tennis problem" (Fellbaum, 1998, p.10).

[4] Initially, we tried collecting web pages in Yahoo! JAPAN. However, we found that most of them were index pages with a few text contents, from which you cannot extract reliable keywords. Though we further tried following links in those index pages to acquire enough texts, extracted words turned out to be site-specific rather than domain-specific since many pages were collected from a particular web site.

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 137–140,
Prague, June 2007. ©2007 Association for Computational Linguistics

Table 1: Examples of Keywords for each Domain

Domain	Examples of Keywords
CULTURE	映画 *movie,* 音楽 *music*
RECREATION	観光 *tourism,* 花火 *firework*
SPORTS	選手 *player,* 野球 *baseball*
HEALTH	手術 *surgery,* 診断 *diagnosis*
LIVING	育児 *childcare,* 家具 *furniture*
DIET	箸 *chopsticks,* 昼食 *lunch*
TRANSPORTATION	駅 *station,* 道路 *road*
EDUCATION	先生 *teacher,* 算数 *arithmetic*
SCIENCE	研究 *research,* 理論 *theory*
BUSINESS	輸入 *import,* 市場 *market*
MEDIA	放送 *broadcast,* 記者 *reporter*
GOVERNMENT	司法 *judicatory,* 税 *tax*

3 Domain Dictionary Construction

To identify which domain a JFW is associated with, we use manually-prepared keywords for each domain rather than document collections. The construction process is as follows: ① Preparing keywords for each domain (§3.1). ② Associating JFWs with domains (§3.2). ③ Reassociating JFWs with NODOMAIN (§3.3). ④ Manual correction (§3.5).

3.1 Preparing Keywords for each Domain

About 20 keywords for each domain were collected manually from words that appear most frequently in the Web. Table 1 shows examples of the keywords.

3.2 Associating JFWs with Domains

A JFW is associated with a domain of the highest A_d score. An A_d score of domain is calculated by summing up the top five A_k scores of the domain. Then, an A_k score, which is defined between a JFW and a keyword of a domain, is a measure that shows how strongly the JFW and the keyword are related (Figure 1). Assuming that two words are related if they cooccur more often than chance in a corpus, we adopt the χ^2 statistics to calculate an A_k score and use web pages as a corpus. The number of co-occurrences is approximated by the number of search engine hits when the two words are used as queries. Among various alternatives, the combination of the χ^2 statistics and web pages is adopted following Sasaki et al. (2006).

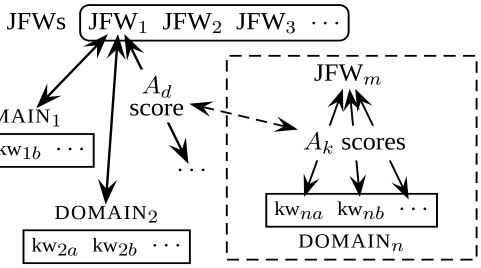


Figure 1: Associating JFWs with Domains

Based on Sasaki et al. (2006), A_k score between a JFW (jw) and a keyword (kw) is given as below.

$$A_k(jw, kw) = \frac{n(ad - bc)^2}{(a+b)(c+d)(a+c)(b+d)}$$

where n is the total number of Japanese web pages,

$$a = hits(jw \mathbin{\&} kw), \quad b = hits(jw) - a,$$
$$c = hits(kw) - a, \quad d = n - (a + b + c).$$

Note that $hits(q)$ represents the number of search engine hits when q is used as a query.

3.3 Reassociating JFWs with NODOMAIN

JFWs that do not belong to any particular domain, i.e. whose highest A_d score is low should be reassociated with NODOMAIN. Thus, a threshold for determining if a JFW's highest A_d score is low is required. The threshold for a JFW (jw) needs to be changed according to $hits(jw)$; the greater $hits(jw)$ is, the higher the threshold should be.

To establish a function that takes jw and returns the appropriate threshold for it, the following semi-automatic process is required after all JFWs are associated with domains: **(i)** Sort all tuples of the form $< jw,\ hits(jw)$, the highest A_d of the $jw >$ by $hits(jw)$.[5] **(ii)** Segment the tuples. **(iii)** For each segment, extract manually tuples whose jw should be associated with one of the 12 domains and those whose jw should be deemed as NODOMAIN. Note that the former tuples usually have higher A_d scores than the latter tuples. **(iv)** For each segment, identify a threshold that distinguishes between the former tuples and the latter tuples by their A_d scores. At this point, pairs of the number of hits (represented by each segment) and the appropriate threshold for it are obtained. **(v)** Approximate the relation between

[5] Note that we acquire the number of search engine hits and the A_d score for each jw in the process ②.

the number of hits and its threshold by a linear function using least-square method. Finally, this function indicates the appropriate threshold for each jw.

3.4 Performance of the Proposed Method

We applied the method to JFWs installed on JUMAN (Kurohashi et al., 1994), which are 26,658 words consisting of commonly used nouns and verbs. As an evaluation, we sampled 380 pairs of a JFW and its domain, and measured accuracy.[6] As a result, the proposed method attained the accuracy of 81.3% (309/380).

3.5 Manual Correction

Our policy is that simpler is better. Thus, as one of our guidelines for manual correction, we avoid associating a JFW with multiple domains as far as possible. JFWs to associate with multiple domains are restricted to those that are EQUALLY relevant to more than one domain.

4 Blog Categorization

As a task-based evaluation, we categorized blog articles into the domains assumed here.

4.1 Categorization Method

(i) Extract JFWs from the article. **(ii)** Classify the extracted JFWs into the domains using the domain dictionary. **(iii)** Sort the domains by the number of JFWs classified in descending order. **(iv)** Categorize the article as the top domain. If the top domain is NODOMAIN, the article is categorized as the second domain under the condition below.

$$|W(\text{2ND DOMAIN})| \div |W(\text{NODOMAIN})| > 0.03$$

where $|W(\text{D})|$ is the number of JFWs classified into the domain D.

4.2 Data

We prepared two blog collections; $B_{controlled}$ and B_{random}. As $B_{controlled}$, 39 blog articles were collected (3 articles for each domain including NODOMAIN) by the following procedure: **(i)** Query the Web using a keyword of the domain.[7] **(ii)** From the top of the search result, collect 3 articles that meet the following conditions; there are enough text contents in it, and people can confidently make a judgment about which domain it is categorized as. As B_{random}, 30 articles were randomly sampled from the Web. Table 2 shows its breakdown.

Table 2: Breakdown of B_{random}

Domain	#	Domain	#
CULTURE	4	DIET	4
RECREATION	1	BUSINESS	12
SPORTS	3	NODOMAIN	5
HEALTH	1		

Note that we manually removed peripheral contents like author profiles or banner advertisements from the articles in both $B_{controlled}$ and B_{random}.

4.3 Result

We measured the accuracy of blog categorization. As a result, the accuracy of 89.7% (35/39) was attained in categorizing $B_{controlled}$, while B_{random} was categorized with 76.6% (23/30) accuracy.

5 Domain Estimation for Unknown Words

We developed an automatic way of estimating the domain of unknown word (uw) using the dictionary.

5.1 Estimation Method

(i) Search the Web by using uw as a query. **(ii)** Retrieve the top 30 documents of the search result. **(iii)** Categorize the documents as one of the domains by the method described in §4.1. **(iv)** Sort the domains by the number of documents in descending order. **(v)** Associate uw with the top domain.

5.2 Experimental Condition

(i) Select 10 words from the domain dictionary for each domain. **(ii)** For each word, estimate its domain by the method in §5.1 after removing the word from the dictionary so that the word is unknown.

5.3 Result

Table 3 shows the number of correctly domain-estimated words (out of 10) for each domain. Accordingly, the total accuracy is 67.5% (81/120).

[6] In the evaluation, one of the authors judged the correctness of each pair.

[7] To collect articles that are categorized as NODOMAIN, we used 日記 *diary* as a query.

Table 3: # of Correctly Domain-estimated Words

Domain	#	Domain	#
CULTURE	7	TRANSPORTATION	7
RECREATION	4	EDUCATION	9
SPORTS	9	SCIENCE	6
HEALTH	9	BUSINESS	9
LIVING	3	MEDIA	2
DIET	7	GOVERNMENT	9

As for the poor accuracy for RECREATION, LIVING, and MEDIA, we found that it was due to either the ambiguous nature of the words of domain or a characteristic of the estimation method. The former brought about the poor accuracy for MEDIA. That is, some words of MEDIA are often used in other contexts. For example, 中継 *live coverage* is often used in the SPORTS context. On the other hand, the method worked poorly for RECREATION and LIVING for the latter reason; the method exploits the Web. Namely, some words of the domains, such as 観光 *tourism* and シャンプー *shampoo*, are often used in the web sites of companies (BUSINESS) that provide services or goods related to RECREATION or LIVING. As a result, the method tends to wrongly associate those words with BUSINESS.

6 Related Work

HowNet (Dong and Dong, 2006) and WordNet provide domain information for Chinese and English, but there has been no domain resource for Japanese that are publicly available.[8]

Domain dictionary construction methods that have been developed so far are all based on highly structured lexical resources like LDOCE or WordNet (Guthrie et al., 1991; Agirre et al., 2001) and hence not applicable to languages for which such highly structured lexical resources are not available.

Accordingly, contributions of this study are twofold: **(i)** We constructed the first Japanese domain dictionary that is fully available. **(ii)** We developed the domain dictionary construction method that requires neither document collections nor highly structured lexical resources.

7 Conclusion

Toward deeper natural language understanding, we constructed the first Japanese domain dictionary that contains 26,658 JFWs. Our method requires neither document collections nor structured lexical resources. The domain dictionary can satisfactorily classify blog articles into the 12 domains assumed in this study. Also, the dictionary can reliably estimate the domain of unknown words except for words that are ambiguous in terms of domains and those that appear frequently in web sites of companies.

Among our future work is to deal with domain information of multiword expressions. For example, 源泉 *fount* and 徴収 *collection* constitute 源泉徴収 *tax deduction at source*. Note that while 源泉 itself belongs to NODOMAIN, 源泉徴収 should be associated with GOVERNMENT.

Also, we will install the domain dictionary on JUMAN (Kurohashi et al., 1994) to make the domain information fully and easily available.

References

Eneko Agirre, Olatz Ansa, David Martinez, and Ed Hovy. 2001. Enriching wordnet concepts with topic signatures. In *Proceedings of the SIGLEX Workshop on "WordNet and Other Lexical Resources: Applications, Extensions, and Customizations" in conjunction with NAACL*.

Zhendong Dong and Qiang Dong. 2006. *HowNet And the Computation of Meaning*. World Scientific Pub Co Inc.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. MIT Press.

Joe A. Guthrie, Louise Guthrie, Yorick Wilks, and Homa Aidinejad. 1991. Subject-Dependent Co-Occurence and Word Sense Disambiguation. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 146–152.

Sadao Kurohashi, Toshihisa Nakamura, Yuji Matsumoto, and Makoto Nagao. 1994. Improvements of Japanese Mophological Analyzer JUMAN. In *Proceedings of the International Workshop on Sharable Natural Language Resources*, pages 22–28.

Yasuhiro Sasaki, Satoshi Sato, and Takehito Utsuro. 2006. Related Term Collection. *Journal of Natural Language Processing*, 13(3):151–176. (in Japanese).

Yumiko Yoshimoto, Satoshi Kinoshita, and Miwako Shimazu. 1997. Processing of proper nouns and use of estimated subject area for web page translation. In *tmi97*, pages 10–18, Santa Fe.

[8] Some human-oriented dictionaries provide domain information. However, domains they cover are all technical ones rather than common domains such as those assumed here.

Extracting Word Sets with Non-Taxonomical Relation

Eiko Yamamoto **Hitoshi Isahara**
Computational Linguistics Group
National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0289, Japan
{eiko, isahara}@nict.go.jp

Abstract

At least two kinds of relations exist among related words: taxonomical relations and thematic relations. Both relations identify related words useful to language understanding and generation, information retrieval, and so on. However, although words with taxonomical relations are easy to identify from linguistic resources such as dictionaries and thesauri, words with thematic relations are difficult to identify because they are rarely maintained in linguistic resources. In this paper, we sought to extract thematically (non-taxonomically) related word sets among words in documents by employing case-marking particles derived from syntactic analysis. We then verified the usefulness of word sets with non-taxonomical relation that seems to be a thematic relation for information retrieval.

1. Introduction

Related word sets are useful linguistic resources for language understanding and generation, information retrieval, and so on. In previous research on natural language processing, many methodologies for extracting various relations from corpora have been developed, such as the "is-a" relation (Hearst 1992), "part-of" relation (Berland and Charniak 1999), causal relation (Girju 2003), and entailment relation (Geffet and Dagan 2005).

Related words can be used to support retrieval in order to lead users to high-quality information. One simple method is to provide additional words related to the key words users have input, such as an input support function within the Google search engine. What kind of relation between the key words that have been input and the additional word is effective for information retrieval?

As for the relations among words, at least two kinds of relations exist: the taxonomical relation and the thematic relation. The former is a relation representing the physical resemblance among objects, which is typically a semantic relation such as a hierarchal, synonymic, or antonymic relation; the latter is a relation between objects through a thematic scene, such as "milk" and "cow" as recollected in the scene "milking a cow," and "milk" and "baby," as recollected in the scene "giving baby milk," which include causal relation and entailment relation. Wisniewski and Bassok (1999) showed that both relations are important in recognizing those objects. However, while taxonomical relations are comparatively easy to identify from linguistic resources such as dictionaries and thesauri, thematic relations are difficult to identify because they are rarely maintained in linguistic resources.

In this paper, we sought to extract word sets with a thematic relation from documents by employing case-marking particles derived from syntactic analysis. We then verified the usefulness of word sets with non-taxonomical relation that seems to be a thematic relation for information retrieval.

2. Method

In order to derive word sets that direct users to obtain information, we applied a method based on the Complementary Similarity Measure (CSM), which can determine a relation between two words in a corpus by estimating inclusive relations between two vectors representing each appearance pattern for each words (Yamamoto *et al.* 2005).

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 141–144,
Prague, June 2007. ©2007 Association for Computational Linguistics

We first extracted word pairs having an inclusive relation between the words by calculating the CSM values. Extracted word pairs are expressed by a tuple <w_i, w_j>, where $CSM(\boldsymbol{V}_i, \boldsymbol{V}_j)$ is greater than $CSM(\boldsymbol{V}_j, \boldsymbol{V}_i)$ when words w_i and w_j have each appearance pattern represented by each binary vector $\boldsymbol{V}_i$ and $\boldsymbol{V}_j$. Then, we connected word pairs with CSM values greater than a certain threshold and constructed word sets. A feature of the CSM-based method is that it can extract not only pairs of related words but also sets of related words because it connects tuples consistently.

Suppose we have <A, B>, <B, C>, <Z, B>, <C, D>, <C, E>, and <C, F> in the order of their CSM values, which are greater than the threshold. For example, let <B, C> be an initial word set {B, C}. First, we find the tuple with the greatest CSM value among the tuples in which the word C at the tail of the current word set is the left word, and connect the right word behind C. In this example, word "D" is connected to {B, C} because <C, D> has the greatest CSM value among the three tuples <C, D>, <C, E>, and <C, F>, making the current word set {B, C, D}. This process is repeated until no tuples exist. Next, we find the tuple with the greatest CSM value among the tuples in which the word B at the head of the current word set is the right word, and connect the left word before B. This process is repeated until no tuples exist. In this example, we obtain the word set {A, B, C, D}.

Finally, we removed ones with a taxonomical relation by using thesaurus. The rest of the word sets have a non-taxonomical relation — including a thematic relation — among the words. We then extracted those word sets that do not agree with the thesaurus as word sets with a thematic relation.

3. Experiment

In our experiment, we used domain-specific Japanese documents within the medical domain (225,402 sentences, 10,144 pages, 37MB) gathered from the Web pages of a medical school and the 2005 Medical Subject Headings (MeSH) thesaurus[1]. Recently, there has been a study on query expansion with this thesaurus as domain information (Friberg 2007).

We extracted word sets by utilizing inclusive relations of the appearance pattern between words based on a modified/modifier relationship in documents. The Japanese language has case-marking particles that indicate the semantic relation between two elements in a dependency relation. Then, we collected from documents dependency relations matching the following five patterns; "A <*no* (of)> B," "P <*wo* (object)> V," "Q <*ga* (subject)> V," "R <*ni* (dative)> V," and "S <*ha* (topic)> V," where A, B, P, Q, R, and S are nouns, V is a verb, and <*X*> is a case-marking particle. From such collected dependency relations, we compiled the following types of experimental data; ***NN-data*** based on co-occurrence between nouns for each sentence, ***NV-data*** based on a dependency relation between noun and verb for each case-marking particle <*wo*>, <*ga*>, <*ni*>, and <*ha*>, and ***SO-data*** based on a collocation between subject and object that depends on the same verb V as the subject. These data are represented with a binary vector which corresponds to the appearance pattern of a noun and these vectors are used as arguments of CSM.

We translated descriptors in the MeSH thesaurus into Japanese and used them as Japanese medical terms. The number of terms appearing in this experiment is 2,557 among them. We constructed word sets consisting of these medical terms. Then, we chose 977 word sets consisting of three or more terms from them, and removed word sets with a taxonomical relation from them with the MeSH thesaurus in order to obtain the rest 847 word sets as word sets with a thematic relation.

4. Verification

In verifying the capability of our word sets to retrieve Web pages, we examined whether they could help limit the search results to more informative Web pages with Google as a search engine.

We assume that addition of suitable key words to the query reduces the number of pages retrieved and the remaining pages are informative pages. Based on this assumption, we examined the decrease of the retrieved pages by additional key words and the contents of the retrieved pages in order to verify the availability of our word sets.

Among 847 word sets, we used 294 word sets in which one of the terms is classified into one category and the rest are classified into another.

[1] The U.S. National Library of Medicine created, maintains, and provides the MeSH® thesaurus.

ovary - spleen - palpation (NN)
variation - cross reactions - outbreaks - secretion (Wo)
bleeding - pyrexia - hematuria - consciousness disorder
- vertigo - high blood pressure (Ga)
space flight - insemination - immunity (Ni)
cough - fetus
- bronchiolitis obliterans organizing pneumonia (Ha)
latency period - erythrocyte - hepatic cell (SO)

Figure 1. Examples of word sets used to verify.

Figure 1 shows examples of the word sets, where terms in a different category are underlined.

In retrieving Web pages for verification, we input the terms composed of these word sets into the search engine. We created three types of search terms from the word set we extracted. Suppose the extracted word set is $\{X_1, ..., X_n, Y\}$, where X_i is classified into one category and Y is classified into another. The first type uses all terms except the one classified into a category different from the others: $\{X_1, ..., X_n\}$ removing Y. The second type uses all terms except the one in the same category as the rest: $\{X_1, ..., X_{k-1}, X_{k+1}, ..., X_n\}$ removing X_k from Type 1. In our experiment, we removed the term X_k with the highest or lowest frequency among X_i. The third type uses terms in Type 2 and Y: $\{X_1, ..., X_{k-1}, X_{k+1}, ..., X_n, Y\}$.

In other words, when we consider the terms in Type 2 as base key words, the terms in Type 1 are key words with the addition of one term having the highest or lowest frequency among the terms in the same category; i.e., the additional term has a feature related to frequency in the documents and is taxonomically related to other terms. The terms in Type 3 are key words with the addition of one term in a category different from those of the other component terms; i.e., the additional term seems to be thematically related — at least non-taxonomically related — to other terms.

First, we quantitatively compared the retrieval results. We used the estimated number of pages retrieved by Google's search engine. Suppose that we first input Type 2 as key words into Google, did not satisfy the result extracted, and added one word to the previous key words. We then sought to determine whether to use Type 1 or Type 3 to obtain more suitable results. The results are shown in Figures 2 and 3, which include the results for the highest frequency and the lowest frequency, respectively. In these figures, the horizontal axis is the number of pages retrieved with Type 2 and the vertical axis is the number of pages retrieved when a certain term is added to Type 2. The circles (•) show the retrieval results with additional key word related taxonomically (Type 1). The crosses (×) show the results with additional key word related non-taxonomically (Type 3). The diagonal line shows that adding one term to the base key words does not affect the number of Web pages retrieved.

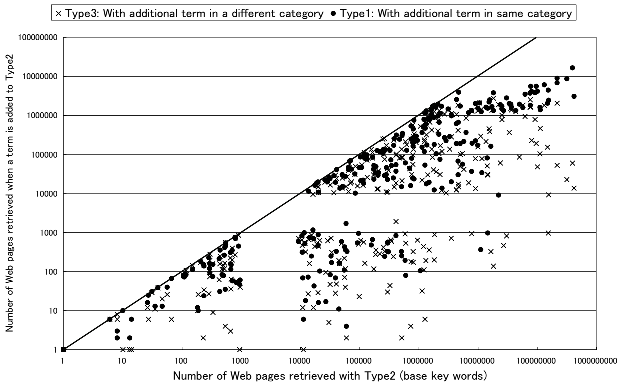


Figure 2. Fluctuation of number of pages retrieved (with the high frequency term).

Type of Data	NN	NV			
		Wo	Ga	Ni	Ha
Word sets for verification	175	43	23	13	26
Cases in which Type 3 defeated Type 1 in retrieval	108	37	15	12	18

Table 1. Number of cases in which Type 3 defeated Type 1 with the high frequency term.

In Figure 2, most crosses fall further below the line. This graph indicates that when searching by Google, adding a search term related non-taxonomically tends to make a bigger difference than adding a term related taxonomically and with high frequency. This means that adding a term related non-taxonomically to the other terms is crucial to retrieving informative pages; that is, such terms are informative terms themselves. Table 1 shows the number of cases in which term in different category decreases the number of hit pages more than high frequency term. By this table, we found that most of the additional terms with high frequency contributed less than additional terms related non-taxonomically to decreasing the number of Web pages retrieved. This means that, in comparison to the high frequency terms, which might not be so informative in themselves, the terms in the other category — related non-taxonomically — are effective for retrieving useful Web pages.

In Figure 3, most circles fall further below the line, in contrast to Figure 2. This indicates that

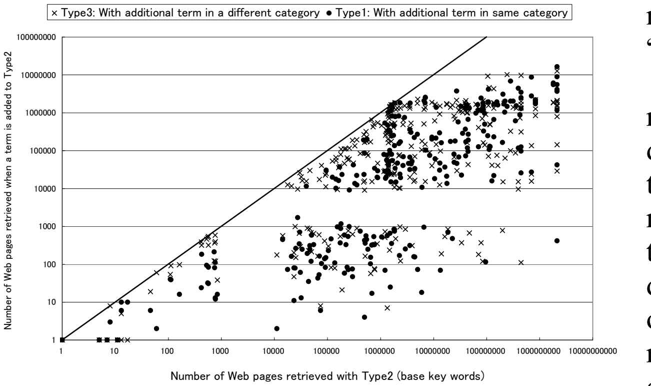


Figure 3. Fluctuation of number of pages retrieved (with the low frequency term).

Type of Data	NN	NV			
		Wo	Ga	Ni	Ha
Word sets for verification	175	43	23	13	26
Cases in which Type 3 defeated Type 1 in retrieval	61	18	7	6	13

Table 2. Number of cases in which Type 3 defeated Type 1 with the low frequency term.

adding a term related taxonomically and with low frequency tends to make a bigger difference than adding a term with high frequency. Certainly, additional terms with low frequency would be informative terms, even though they are related taxonomically, because they may be rare terms on the Web and therefore the number of pages containing the term would be small. Table 2 shows the number of cases in which term in different category decreases the number of hit pages more than low frequency term. In comparing these numbers, we found that the additional term with low frequency helped to reduce the number of Web pages retrieved, making no effort to determine the kind of relation the term had with the other terms. Thus, the terms with low frequencies are quantitatively effective when used for retrieval. However, if we compare the results retrieved with Type 1 search terms and Type 3 search terms, it is clear that big differences exist between them.

For example, consider "latency period - erythrocyte - hepatic cell" obtained from SO-data in Figure 1. "Latency period" is classified into a category different from the other terms and "hepatic cell" has the lowest frequency in this word set. When we used all the three terms, we obtained pages related to "malaria" at the top of the results and the title of the top page was "What is malaria?" in Japanese. With "latency period" and "erythrocyte," we again obtained the same page at the top, although it was not at the top when we used "erythrocyte" and "hepatic cell" which have a taxonomical relation.

As we showed above, the terms with thematic relations with other search terms are effective at directing users to informative pages. Quantitatively, terms with a high frequency are not effective at reducing the number of pages retrieved; qualitatively, low frequency terms may not effective to direct users to informative pages. We will continue our research in order to extract terms in thematic relation more accurately and verify the usefulness of them more quantitatively and qualitatively.

5. Conclusion

We sought to extract word sets with a thematic relation from documents by employing case-marking particles derived from syntactic analysis. We compared the results retrieved with terms related only taxonomically and the results retrieved with terms that included a term related non-taxonomically to the other terms. As a result, we found adding term which is thematically related to terms that have already been input as key words is effective at retrieving informative pages.

References

Berland, M. and Charniak, E. 1999. Finding parts in very large corpora, In *Proceedings of ACL 99*, 57–64.

Friberg, K. 2007. Query expansion using domain information in compounds, In *Proceedings of NAACL-HLT 2007 Doctoral Consortium*, 1–4.

Geffet, M. and Dagan, I. 2005. The distribution inclusion hypotheses and lexical entailment. In *Proceedings of ACL 2005*, 107–114.

Girju, R. 2003. Automatic detection of causal relations for question answering. In *Proceedings of ACL Workshop on Multilingual summarization and question answering*, 76–114.

Hearst, M. A. 1992, Automatic acquisition of hyponyms from large text corpora, In *Proceedings of Coling 92*, 539–545.

Wisniewski, E. J. and Bassok. M. 1999. What makes a man similar to a tie? *Cognitive Psychology*, 39: 208–238.

Yamamoto, E., Kanzaki, K., and Isahara, H. 2005. Extraction of hierarchies based on inclusion of co-occurring words with frequency information. In *Proceedings of IJCAI 2005*, 1166–1172.

A Linguistic Service Ontology for Language Infrastructures

Yoshihiko Hayashi
Graduate School of Language and Culture, Osaka University
1-8 Machikaneyama-cho, Toyonaka, 560-0043 Japan
hayashi@lang.osaka-u.ac.jp

Abstract

This paper introduces conceptual framework of an ontology for describing linguistic services on network-based language infrastructures. The ontology defines a taxonomy of processing resources and the associated static language resources. It also develops a sub-ontology for abstract linguistic objects such as expression, meaning, and description; these help define functionalities of a linguistic service. The proposed ontology is expected to serve as a solid basis for the interoperability of technical elements in language infrastructures.

1 Introduction

Several types of linguistic services are currently available on the Web, including text translation and dictionary access. A variety of NLP tools is also available and public. In addition to these, a number of community-based language resources targeting particular domains of application have been developed, and some of them are ready for dissemination. A composite linguistic service tailored to a particular user's requirements would be composable, if there were a language infrastructure on which elemental linguistic services, such as NLP tools, and associated language resources could be efficiently combined. Such an infrastructure should provide an efficient mechanism for creating workflows of composite services by means of authoring tools for the moment, and through an automated planning in the future.

To this end, technical components in an infrastructure must be properly described, and the semantics of the descriptions should be defined based on a shared ontology.

2 Architecture of a Language Infrastructure

The linguistic service ontology described in this paper has not been intended for a particular language infrastructure. However we expect that the ontology should be first introduced in an infrastructure like the Language Grid[1], because it, unlike other research-oriented infrastructures, tries to incorporate a wide range of NLP tools and community-based language resources (Ishida, 2006) in order to be useful for a range of intercultural collaboration activities.

The fundamental technical components in the Language Grid could be: (a) external web-based services, (b) on-site NLP core functions, (c) static language resources, and (d) wrapper programs.

Figure 1 depicts the general architecture of the infrastructure. The technical components listed above are deployed as shown in the figure.

Computational nodes in the language grid are classified into the following two types as described in (Murakami et al., 2006).

- A *service node* accommodates atomic linguistic services that provide functionalities of the NLP tool/system running on a node, or they can simply have a wrapper program that consults an external web-based linguistic service.
- A *core node* maintains a repository of the known atomic linguistic services, and provides service discovery functionality to the possible users/applications. It also maintains a workflow re-

[1] Language Grid: http://langrid.nict.go.jp/

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 145–148,
Prague, June 2007. ©2007 Association for Computational Linguistics

pository for composite linguistic services, and is equipped with a workflow engine.

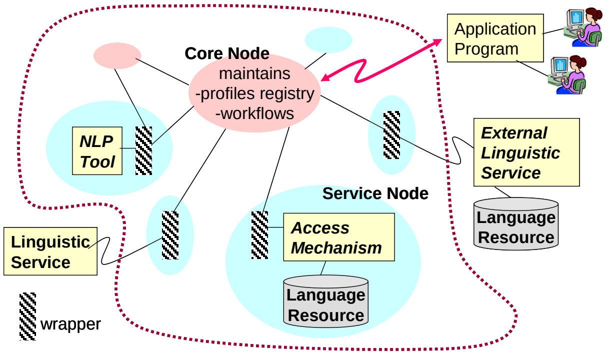


Figure 1. Architecture of a Language Infrastructure.

Given a technical architecture like this, the linguistic service ontology will serve as a basis for composition of composite linguistic services, and efficient wrapper generation. The wrapper generation processes are unavoidable during incorporation of existing general linguistic services or dissemination of newly created community-based language resources. Tthe most important desideratum for the ontology, therefore, is that it be able to specify the input/output constraints of a linguistic service properly. Such input/output specifications enable us to derive a taxonomy of linguistic service and the associated language resources.

3 The Upper Ontology

3.1 The top level

We have developed the upper part of the service ontology so far, and have been working on detailing some of its core parts. Figure 2 shows the top level of the proposed linguistic service ontology.

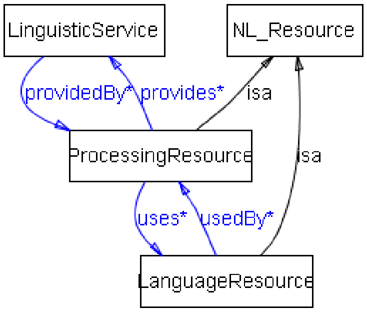


Figure 2. The Top Level of the Ontology.

The topmost class is **NL_Resource**, which is partitioned into **ProcessingResource**, and **LanguageResource**. Here, as in GATE (Cunningham, 2002), processing resource refers to programmatic or algorithmic resources, while language resource refers to data-only static resources such as lexicons or corpora. The innate relation between these two classes is: a processing resource can use language resources. This relationship is specifically introduced to properly define linguistic services that are intended to provide access functions to language resources.

As shown in the figure, **LinguisticService** is *provided by* a processing resource, stressing that any linguistic service is realized by a processing resource, even if its prominent functionality is accessing language resources in response to a user's query. It also has the meta-information for advertising its non-functional descriptions.

The fundamental classes for abstract linguistic objects, **Expression**, **Meaning**, and **Description** and the innate relations among them are illustrated in Figure 3. These play roles in defining functionalities of some types of processing resources and associated language resources. As shown in Fig. 3, an expression may *denote* a meaning, and the meaning can be further *described by* a description, especially for human uses.

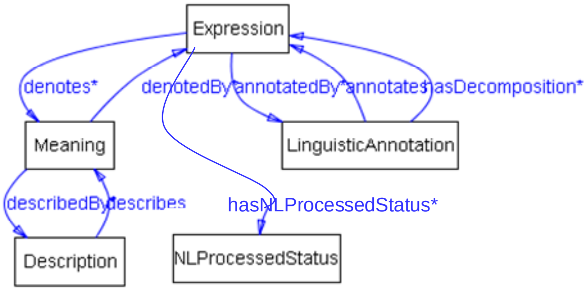


Figure 3. Classes for Abstract Linguistic Objects.

In addition to these, **NLProcessedStatus** and **LinguisticAnnotation** are important in the sense that NLP status represents the so-called IOPE (Input-Output-Precondition-Effect) parameters of a linguistic processor, which is a subclass of the processing resource, and the data schema for the results of a linguistic analysis is defined by using the linguistic annotation class.

3.2 Taxonomy of language resources

The language resource class currently is partitioned into subclasses for **Corpus** and **Dictionary**. The immediate subclasses of the dictionary class are: (1) **MonolingualDictionary**, (2) **Bi-**

lingualDictionary, (3) **MultilingualTerminology**, and (4) **ConceptLexicon**. The major instances of (1) and (2) are so-called machine-readable dictionaries (MRDs). Many of the community-based special language resources should fall into (3), including multilingual terminology lists specialized for some application domains. For subclass (4), we consider the computational concept lexicons, which can be modeled by a WordNet-like encoding framework (Hayashi and Ishida, 2006).

3.3 Taxonomy of processing resources

The top level of the processing resource class consists of the following four subclasses, which take into account the input/output constraints of processing resources, as well as the language resources they utilize.

- **AbstractReader**, **AbstractWriter**: These classes are introduced to describe computational processes that convert to-and-from non-textual representation (e.g. speech) and textual representation (character strings).
- **LR_Accessor**: This class is introduced to describe language resource access functionalities. It is first partitioned into **CorpusAccessor** and **DictionaryAccessor**, depending on the type of language resource it accesses. The input to a language resource accessor is a query (**LR_AccessQuery**, sub-class of **Expression**), and the output is a kind of 'dictionary meaning' (**DictionaryMeaning**), which is a sub-class of meaning class. The dictionary meaning class is further divided into sub-classes by referring to the taxonomy of dictionary.
- **LinguisticProcessor**: This class is further discussed in the next subsection.

3.4 Linguistic processors

The linguistic processor class is introduced to represent NLP tools/systems. Currently and tentatively, the linguistic processor class is first partitioned into **Transformer** and **Analyzer**.

The transformer class is introduced to represent **Paraphrasor** and **Translator**; both rewrite the input linguistic expression into another expression while maintaining the original meaning. The only difference is the sameness of the input/output languages. We explicitly express the input/output language constraints in each class definition.

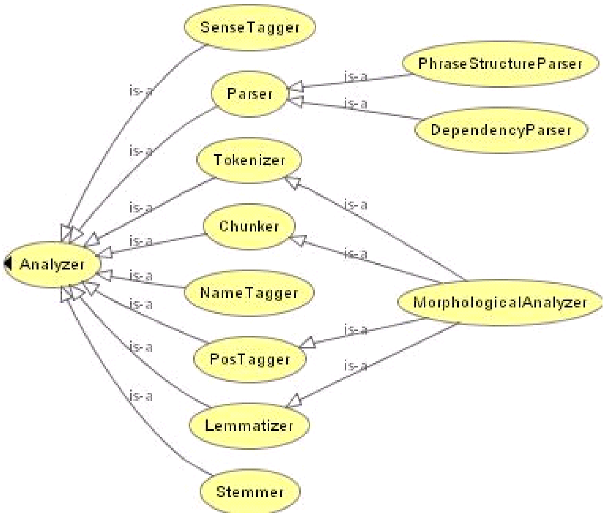


Figure 4. Taxonomy of Linguistic Analyzer.

Figure 4 shows the working taxonomy of the analyzer class. While it is not depicted in the figure, the input/output constraints of a linguistic analyzer are specified by the **Expression** class, while its precondition/effect parameters are defined by **NLProcessedStatus** class. The details are also not shown in this figure, these constraints are further restricted with respect to the taxonomy of the processing resource.

We also assume that any linguistic analyzer additively annotates some linguistic information to the input, as proposed by (Cunningham, 2002), (Klein and Potter, 2004). That is, an analyzer working at a certain linguistic level (or 'depth') adds the corresponding level of annotations to the input. In this sense, any natural language expression can have a layered/multiple linguistic annotation. To make this happen, a linguistic service ontology has to appropriately define a sub-ontology for the linguistic annotations by itself or by incorporating some external standard, such as LAF (Ide and Romary, 2004).

3.5 NLP status and the associated issues

Figure 5 illustrates our working taxonomy of NLP processed status. Note that, in this figure, only the portion related to linguistic analyzer is detailed. Benefits from the NLP status class will be twofold: (1) as a part of the description of a linguistic analyzer, we assign corresponding instances of this class as its precondition/effect parameters, (2) any instance of the expression class can be concisely 'tagged' by instances of the NLP status class, according to how 'deeply' the expression has been linguistically analyzed so far. Essentially, such information can be retrieved from the attached linguistic annotations. In this sense, the NLP status class might be redundant. Tagging an instance of expression in that way, however, can be reasonable: we can define the input/output constraints of a linguistic analyzer concisely with this device.

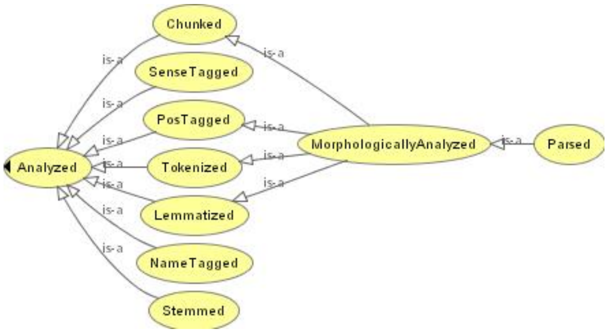


Figure 5. Taxonomy of NLP Status.

Each subclass in the taxonomy represents the type or level of a linguistic analysis, and the hierarchy depicts the processing constraints among them. For example, if an expression has been parsed, it would already have been morphologically analyzed, because parsing usually requires the input to be morphologically analyzed beforehand. The subsumption relations encoded in the taxonomy allow simple reasoning in possible composite service composition processes. However note that the taxonomy is only preliminary. The arrangement of the subclasses within the hierarchy may end up being far different, depending on the languages considered, and the actual NLP tools, these are essentially idiosyncratic, that are at hand. For example, the notion of 'chunk' may be different from language to language. Despite of these, if we go too far in this direction, constructing a taxonomy would be meaningless, and we would forfeit reasonable generalities.

4 Related Works

Klein and Potter (2004) have once proposed an ontology for NLP services with OWL-S definitions. Their proposal however has not included detailed taxonomies either for language resources, or for abstract linguistic objects, as shown in this paper. Graça, et al. (2006) introduced a framework for integrating NLP tools with a client-server architecture having a multi-layered repository. They also proposed a data model for encoding various types of linguistic information. However the model itself is not ontologized as proposed in this paper.

5 Concluding Remarks

Although the proposed ontology successfully defined a number of first class objects and the innate relations among them, it must be further refined by looking at specific NLP tools/systems and the associated language resources. Furthermore, its effectiveness in composition of composite linguistic services or wrapper generation should be demonstrated on a specific language infrastructure such as the Language Grid.

Acknowledgments

The presented work has been partly supported by NICT international joint research grant. The author would like to thank to Thierry Declerck and Paul Buitelaar (DFKI GmbH, Germany) for their helpful discussions.

References

H. Cunningham, et al. 2002. GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. *Proc. of ACL 2002*, pp.168-175.

J. Graça , et al. 2006. NLP Tools Integration Using a Multi-Layered Repository. P*roc. of LREC 2006 Workshop on Merging and Layering Linguistic Information.*

Y. Hayashi and T. Ishida. 2006. A Dictionary Model for Unifying Machine Readable Dictionaries and Computational Concept Lexicons. *Proc. of LREC 2006*, pp.1-6.

N. Ide and L. Romary. 2004. International Standard for a Linguistic Annotation Framework. *Journal of Natural Language Engineering*, Vol.10:3-4, pp.211-225.

T. Ishida. 2006. Language Grid: An Infrastructure for Intercultural Collaboration. *Proc.* of *SAINT-06*, pp. 96-100, keynote address.

E. Klein and S. Potter. 2004. An Ontology for NLP Services. *Proc. of LREC 2004 Workshop on Registry of Linguistic Data Categories.*

Y. Murakami, et al. 2006. Infrastructure for Language Service Composition. *Proc. of Second International Conference on Semantics, Knowledge, Grid.*

Empirical Measurements of Lexical Similarity in Noun Phrase Conjuncts

Deirdre Hogan[*]
Department of Computer Science
Trinity College Dublin
Dublin 2, Ireland
dhogan@computing.dcu.ie

Abstract

The ability to detect similarity in conjunct heads is potentially a useful tool in helping to disambiguate coordination structures - a difficult task for parsers. We propose a distributional measure of similarity designed for such a task. We then compare several different measures of word similarity by testing whether they can empirically detect similarity in the head nouns of noun phrase conjuncts in the Wall Street Journal (WSJ) treebank. We demonstrate that several measures of word similarity can successfully detect conjunct head similarity and suggest that the measure proposed in this paper is the most appropriate for this task.

1 Introduction

Some noun pairs are more likely to be conjoined than others. Take the follow two alternate bracketings: 1. *busloads of ((executives) and (their spouses))* and 2. *((busloads of executives) and (their spouses))*. The two head nouns coordinated in 1 are *executives* and *spouses*, and (incorrectly) in 2: *busloads* and *spouses*. Clearly, the former pair of head nouns is more likely and, for the purpose of discrimination, a parsing model would benefit if it could learn that *executives and spouses* is a more likely combination than *busloads and spouses*. If nouns co-occurring in coordination patterns are often semantically similar, and if a similarity measure could be defined so that, for example: $sim(executives, spouses) > sim(busloads, spouses)$ then it is potentially useful for coordination disambiguation.

The idea that nouns co-occurring in conjunctions tend to be semantically related has been noted in (Riloff and Shepherd, 1997) and used effectively to automatically cluster semantically similar words (Roark and Charniak, 1998; Caraballo, 1999; Widdows and Dorow, 2002). The tendency for conjoined nouns to be semantically similar has also been exploited for coordinate noun phrase disambiguation by Resnik (1999) who employed a measure of similarity based on WordNet to measure which were the head nouns being conjoined in certain types of coordinate noun phrase.

In this paper we look at different measures of word similarity in order to discover whether they can detect empirically a tendency for conjoined nouns to be more similar than nouns which co-occur but are not conjoined. In Section 2 we introduce a measure of word similarity based on word vectors and in Section 3 we briefly describe some WordNet similarity measures which, in addition to our word vector measure, will be tested in the experiments of Section 4.

2 Similarity based on Coordination Co-occurrences

The potential usefulness of a similarity measure depends on the particular application. An obvious place to start, when looking at similarity functions for measuring the type of semantic similarity common for coordinate nouns, is a similarity function based on distributional similarity with context de-

[*] Now at the National Centre for Language Technology, Dublin City University, Ireland.

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 149–152,
Prague, June 2007. ©2007 Association for Computational Linguistics

fined in terms of coordination patterns. Our measure of similarity is based on noun co-occurrence information, extracted from conjunctions and lists. We collected co-occurrence data on $82,579$ distinct word types from the BNC and the WSJ treebank.

We extracted all noun pairs from the BNC which occurred in a pattern of the form: *noun cc noun*[1], as well as lists of any number of nouns separated by commas and ending in *cc noun*. Each noun in the list is linked with every other noun in the list. Thus for a list: n_1, n_2, and n_3, there will be co-occurrences between words n_1 and n_2, between n_1 and n_3 and between n_2 and n_3. To the BNC data we added all head noun pairs from the WSJ (sections 02 to 21) that occurred together in a coordinate noun phrase.[2]

From the co-occurrence data we constructed word vectors. Every dimension of a word vector represents another word type and the values of the components of the vector, the term weights, are derived from the coordinate word co-occurrence counts. We used dampened co-occurrence counts, of the form: $1 + log(count)$, as the term weights for the word vectors. To measure the similarity of two words, w_1 and w_2, we calculate the cosine of the angle between the two word vectors, $\vec{w}_1$ and $\vec{w}_2$.

3 WordNet-Based Similarity Measures

We also examine the following measures of semantic similarity which are WordNet-based.[3] Wu and Palmer (1994) propose a measure of similarity of two concepts c_1 and c_2 based on the depth of concepts in the WordNet hierarchy. Similarity is measured from the depth of the most specific node dominating both c_1 and c_2, (their lowest common subsumer), and normalised by the depths of c_1 and c_2. In (Resnik, 1995) concepts in WordNet are augmented by corpus statistics and an information-theoretic measure of semantic similarity is calculated. Similarity of two concepts is measured by the information content of their lowest common subsumer in the *is-a* hierarchy of WordNet. Both Jiang and Conrath (1997) and Lin (1998) propose extentions of Resnik's measure. Leacock and Chodorow (1998)'s measure takes into account the path length between two concepts, which is scaled by the depth of the hierarchy in which they reside. In (Hirst and St-Onge, 1998) similarity is based on path length as well as the number of changes in the direction in the path. In (Banerjee and Pedersen, 2003) semantic relatedness between two concepts is based on the number of shared words in their WordNet definitions (glosses). The gloss of a particular concept is extended to include the glosses of other concepts to which it is related in the WordNet hierarchy. Finally, Patwardhan and Pederson (2006) build on previous work on second-order co-occurrence vectors (Schütze, 1998) by constructing second-order co-occurrence vectors from WordNet glosses, where, as in (Banerjee and Pedersen, 2003), the gloss of a concept is extended so that it includes the gloss of concepts to which it is directly related in WordNet.

4 Experiments

We selected two sets of data from sections 00, 01, 22 and 24 of the WSJ treebank. The first consists of all nouns pairs which make up the head words of two conjuncts in coordinate noun phrases (again not including coordinate NPBs). We found 601 such coordinate noun pairs. The second data set consists of 601 word pairs which were selected at random from all head-modifier pairs where both head and modifier words are nouns and are *not* coordinated. We tested the 9 different measures of word similarity just described on each data set in order to see if a significant difference could be detected between the similarity scores for the coordinate words sample and non-coordinate words sample.

Initially both the coordinate and non-coordinate pair samples each contained 601 word pairs. However, before running the experiments we removed all pairs where the words in the pair were identical. This is because identical words occur more often in coordinate head words than in other lexical dependencies (there were 43 pairs with identical words in the coordination set, compared to 3 such pairs in the

[1] It would be preferable to ensure that the pairs extracted are unambiguously conjoined heads. We leave this to future work.

[2] We did not include coordinate head nouns from base noun phrases (NPB) (i.e. noun phrases that do not dominate other noun phrases) because the underspecified annotation of NPBs in the WSJ means that the conjoined head nouns can not always be easily identified.

[3] All of the WordNet-based similarity measure experiments, as well as a random similarity measure, were carried out with the WordNet::Similarity package, http://search.cpan.org/dist/WordNet-Similarity.

SimTest	n_{coord}	$\overline{x}_{coord}$	SD_{coord}	$n_{nonCoord}$	$\overline{x}_{nonCoord}$	$SD_{nonCoord}$	95% CI	p-value
coordDistrib	503	0.11	0.13	485	0.06	0.09	[0.04 0.07]	0.000
(Resnik, 1995)	444	3.19	2.33	396	2.43	2.10	[0.46 1.06]	0.000
(Lin, 1998)	444	0.27	0.26	396	0.19	0.22	[0.04 0.11]	0.000
(Jiang and Conrath, 1997)	444	0.13	0.65	395	0.07	0.08	[-0.01 0.11]	0.083
(Wu and Palmer, 1994)	444	0.63	0.19	396	0.55	0.19	[0.06 0.11]	0.000
(Leacock and Chodorow, 1998)	444	1.72	0.51	396	1.52	0.47	[0.13 0.27]	0.000
(Hirst and St-Onge, 1998)	459	1.599	2.03	447	1.09	1.87	[0.25 0.76]	0.000
(Banerjee and Pedersen, 2003)	451	114.12	317.18	436	82.20	168.21	[-1.08 64.92]	0.058
(Patwardhan and Pedersen, 2006)	459	0.67	0.18	447	0.66	0.2	[-0.02 0.03]	0.545
random	483	0.89	0.17	447	0.88	0.18	[-0.02 0.02]	0.859

Table 1: Summary statistics for 9 different word similarity measures (plus one random measure):n_{coord} and $n_{nonCoord}$ are the sample sizes for the coordinate and non-coordinate noun pairs samples, respectively; $\overline{x}_{coord}$, SD_{coord} and $\overline{x}_{nonCoord}$, $SD_{nonCoord}$ are the sample means and standard deviations for the two sets. The 95% CI column shows the 95% confidence interval for the difference between the two sample means. The p-value is for a Welch two sample two-sided t-test. *coordDistrib* is the measure introduced in Section 2.

non-coordination set). If we had not removed them, a statistically significant difference between the similarity scores of the pairs in the two sets could be found simply by using a measure which, say, gave one score for identical words and another (lower) score for all non-identical word pairs.

Results for all similarity measure tests on the data sets described above are displayed in Table 1. In one final experiment we used a random measure of similarity. For each experiment we produced two samples, one consisting of the similarity scores given by the similarity measure for the coordinate noun pairs, and another set of similarity scores generated for the non-coordinate pairs. The sample sizes, means, and standard deviations for each experiment are shown in the table. Note that the variation in the sample size is due to coverage: the different measures did not produce a score for all word pairs. Also displayed in Table 1 are the results of statistical significance tests based on the Welsh two sample t-test. A 95% confidence interval for the difference of the sample means is shown along with the p-value.

5 Discussion

For all but three of the experiments (excluding the random measure), the difference between the mean similarity measures is statistically significant. Interestingly, the three tests where no significant difference was measured between the scores on the coordination set and the non-coordination set (Jiang and Conrath, 1997; Banerjee and Pedersen, 2003; Patwardhan and Pedersen, 2006) were the three top scoring measures in (Patwardhan and Pedersen, 2006), where a subset of six of the above WordNet-based experiments were compared and the measures evaluated against human relatedness judgements and in a word sense disambiguation task. In another comparative study (Budanitsky and Hirst, 2002) of five of the above WordNet-based measures, evaluated as part of a real-word spelling correction system, Jiang and Conrath (1997)'s similarity score performed best. Although performing relatively well under other evaluation criteria, these three measures seem less suited to measuring the kind of similarity occurring in coordinate noun pairs. One possible explanation for the unsuitability of the measures of (Patwardhan and Pedersen, 2006) for the coordinate similarity task could be based on how context is defined when building context vectors. Context for an instance of the the word *w* is taken to be the words that surround *w* in the corpus within a given number of positions, where the corpus is taken as all the glosses in WordNet. Words that form part of collocations such as *disk drives* or *task force* would then tend to have very similar contexts, and thus such word pairs, from non-coordinate modifier-head relations, could be given too high a similarity score.

Although the difference between the mean similarity scores seems rather slight in all experiments, it is worth noting that not all coordinate head words *are* semantically related. To take a couple of examples from the coordinate word pair set: *work/harmony* extracted from *hard work and harmony*, and *power/clause* extracted from *executive power and the appropriations clause*. We would not expect these word pairs to get a high similarity score. On the other hand, it is also possible that some of the examples of non-coordinate dependencies involve semantically similar words. For example, nouns in lists are often semantically similar, and we did not exclude nouns extracted from lists from the non-coordinate test set.

Although not all coordinate noun pairs are semantically similar, it seems clear, on inspection of the two sets of data, that they are more likely to be semantically similar than modifier-head word pairs, and the tests carried out for most of the measures of semantic similarity detect a significant difference between the similarity scores assigned to coordinate pairs and those assigned to non-coordinate pairs.

It is not possible to judge, based on the significance tests alone, which might be the most useful measure for the purpose of disambiguation. However, in terms of coverage, the distributional measure introduced in Section 2 clearly performs best[4]. This measure of distributional similarity is perhaps more suited to the task of coordination disambiguation because it directly measures the type of similarity that occurs between coordinate nouns. That is, the distributional similarity measure presented in Section 2 defines two words as similar if they occur in coordination patterns with a similar set of words and with similar distributions. Whether the words are *semantically* similar becomes irrelevant. A measure of semantic similarity, on the other hand, might find words similar which are quite unlikely to appear in coordination patterns. For example, Cederberg and Widdows (2003) note that words appearing in coordination patterns tend to be on the same ontological level: ‘fruit and vegetables’ is quite likely to occur, whereas ‘fruit and apples’ is an unlikely co-occurrence. A WordNet-based measure of semantic similarity, however, might give a high score to both of the noun pairs.

In the future we intend to use the similarity measure outlined in Section 2 in a lexicalised parser to help resolve coordinate noun phrase ambiguities.

Acknowledgements Thanks to the TCD Broad Curriculum Fellowship and to the SFI Research Grant 04/BR/CS370 for funding this research. Thanks also to Pádraig Cunningham, Saturnino Luz and Jennifer Foster for helpful discussions.

[4] Somewhat unsurprisingly given it is part trained on data from the same domain.

References

Satanjeev Banerjee and Ted Pedersen. 2003 Extended Gloss Overlaps as a Measure of Semantic Relatedness. In *Proceeding of the 18th IJCAI.*

Alexander Budanitsky and Graeme Hirst. 2002 Semantic Distance in WordNet: An experimental, application-oriented Evaluation of Five Measures In *Proceedings of the 3rd CICLING.*

Sharon Caraballo. 1999 Automatic construction of a hypernym-labeled noun hierarchy from text In *Proceedings of the 37th ACL.*

Scott Cederberg and Dominic Widdows. 2003. Using LSA and Noun Coordination Information to Improve the Precision and Recall of Automatic Hyponymy Extraction. In *Proceedings of the 7th CoNLL.*

G. Hirst and D. St-Onge 1998. Lexical Chains as representations of context for the detection and correction of malapropisms. *WordNet: An electronic lexical database.* MIT Press.

J. Jiang and D. Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *Proceedings of the ROCLING.*

C. Leacock and M. Chodorow. 1998. Combining local context and WordNet similarity for word sense identification. *WordNet: An electronic lexical database.* MIT Press.

D. Lin. 1998. An information-theoretic definition of similarity. In *Proceedings of the 15th ICML.*

Siddharth Patwardhan and Ted Pedersen. 2006. Using WordNet-based Context Vectors to Estimate the Semantic Relatedness of Concepts. In *Proceedings of Making Sense of Sense - Bringing Computational Linguistics and Psycholinguistics Together, EACL.*

Philip Resnik. 1995. Using Information Content to Evaluate Semantic Similarity. In *Proceedings of IJCAI.*

Philip Resnik. 1999. Semantic Similarity in a Taxonomy: An Information-Based Measure and its Application to Problems of Ambiguity in Natural Language. In *Journal of Artificial Intelligence Research*, 11:95-130.

Ellen Riloff and Jessica Shepherd 1997. A Corpus-based Approach for Building Semantic Lexicon. In *Proceedings of the 2nd EMNLP.*

Brian Roark and Eugene Charniak 1998. Noun-phrase Co-occurrence Statistics for Semi-automatic semantic lexicon construction. In *Proceedings of the COLING-ACL.*

Hinrich Schütze. 1998. Automatic Word Sense Discrimination. *Computational Linguistics*, 24(1):97-123.

Dominic Widdows and Beate Dorow. 2002. A Graph Model for Unsupervised Lexical Acquisition. In *Proceedings of the 19th COLING.*

Zhibiao Wu and Martha Palmer. 1994. Verb Semantics and Lexical Selection. In *Proceedings of the ACL.*

Automatic Discovery of Named Entity Variants – Grammar-driven Approaches to Non-alphabetical Transliterations

Chu-Ren Huang
Institute of Linguistics
Academia Sinica, Taiwan
churenhuang@gmail.com

Petr Šimon
Institute of Linguistics
Academia Sinica, Taiwan
sim@klubko.net

Shu-Kai Hsieh
DoFLAL
NIU, Taiwan
shukai@gmail.com

Abstract

Identification of transliterated names is a particularly difficult task of Named Entity Recognition (NER), especially in the Chinese context. Of all possible variations of transliterated named entities, the difference between PRC and Taiwan is the most prevalent and most challenging. In this paper, we introduce a novel approach to the automatic extraction of diverging transliterations of foreign named entities by bootstrapping co-occurrence statistics from tagged and segmented Chinese corpus. Preliminary experiment yields promising results and shows its potential in NLP applications.

1 Introduction

Named Entity Recognition (NER) is one of the most difficult problems in NLP and Document Understanding. In the field of Chinese NER, several approaches have been proposed to recognize personal names, date/time expressions, monetary and percentage expressions. However, the discovery of transliteration variations has not been well-studied in Chinese NER. This is perhaps due to the fact that the transliteration forms in a non-alphabetic language such as Chinese are opaque and not easy to compare. On the hand, there is often more than one way to transliterate a foreign name. On the other hand, dialectal difference as well as different transliteration strategies often lead to the same named entity to be transliterated differently in different Chinese speaking communities.

Corpus	Example (Clinton)	Frequency
XIN	克林頓	24382
CNA	克林頓	150
XIN	柯林頓	0
CNA	柯林頓	120842

Table 1: Distribution of two transliteration variants for ”Clinton” in two sub-corpora

Of all possible variations, the cross-strait difference between PRC and Taiwan is the most prevalent and most challenging.[1]The main reason may lie in the lack of suitable corpus.

Even given some subcorpora of PRC and Taiwan variants of Chinese, a simple contrastive approach is still not possible. It is because: (1) some variants might overlap and (2) there are more variants used in each corpus due to citations or borrowing cross-strait. Table 1 illustrates this phenomenon, where CNA stands for Central News Agency in Taiwan, XIN stands for Xinhua News Agency in PRC, respectively.

With the availability of Chinese Gigaword Corpus (CGC) and Word Sketch Engine (WSE) Tools (Kilgarriff, 2004). We propose a novel approach towards discovery of transliteration variants by utilizing a full range of grammatical information augmented with phonological analysis.

Existing literatures on processing of transliteration concentrate on the identification of either the transliterated term or the original term, given knowledge of the other (e.g. (Virga and Khudanpur,

[1] For instance, we found at least 14 transliteration variants for Lewinsky,such as 呂茵斯基，呂文絲基，呂茵斯，陸文斯基，陸茵斯 基，柳思基，陸雯絲姬，陸文斯基，呂茵斯基，露文斯基，李文斯基，露温 斯基，蘿恩斯基，李雯斯基 and so on.

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 153–156,
Prague, June 2007.

2003)). These studies are typically either rule-based or statistics-based, and specific to a language pair with a fixed direction (e.g. (Wan and Verspoor, 1998; Jiang et al., 2007)). To the best of our knowledge, ours is the first attempt to discover transliterated NE's without assuming prior knowledge of the entities. In particular, we propose that transliteration variants can be discovered by extracting and comparing terms from similar linguistic context based on CGC and WSE tools. This proposal has great potential of increasing robustness of future NER work by enabling discovery of new and unknown transliterated NE's.

Our study shows that resolution of transliterated NE variations can be fully automated. This will have strong and positive implications for cross-lingual and multi-lingual informational retrieval.

2 Bootstrapping transliteration pairs

The current study is based on Chinese Gigaword Corpus (CGC) (Graff el al., 2005), a large corpus contains with 1.1 billion Chinese characters containing data from Central News Agency of Taiwan (ca. 700 million characters), Xinhua News Agency of PRC (ca. 400 million characters). These two sub-corpora represent news dispatches from roughly the same period of time, i.e. 1990-2002. Hence the two sub-corpora can be expected to have reasonably parallel contents for comparative studies.[2]

The premises of our proposal are that transliterated NE's are likely to collocate with other transliterated NE's, and that collocates of a pair of transliteration variants may form contrasting pairs and are potential variants. In particular, since the transliteration variations that we are interested in are those between PRC and Taiwan Mandarin, we will start with known contrasting pairs of these two language variants and mine potential variant pairs from their collocates. These potential variant pairs are then checked for their phonological similarity to determine whether they are true variants or not. In order to effectively select collocates from specific grammatical constructions, the Chinese Word Sketch[3] is adopted. In particular, we use the Word Sketch difference (WSDiff) function to pick the grammatical contexts as well as contrasting pairs. It is important to bear in mind that Chinese texts are composed of Chinese characters, hence it is impossible to compare a transliterated NE with the alphabetical form in its original language. The following characteristics of a transliterated NE's in CGC are exploited to allow discovery of transliteration variations without referring to original NE.

- *frequent co-occurrence of named entities within certain syntagmatic relations* – named entities frequently co-occur in relations such as AND or OR and this fact can be used to collect and score mutual predictability.

- *foreign named entities are typically transliterated phonetically* – transliterations of the same name entity using different characters can be matched by using simple heuristics to map their phonological value.

- *presence and co-occurrence of named entities in a text is dependent on a text type* – journalistic style cumulates many foreign named entities in close relations.

- *many entities will occur in different domains* – famous person can be mentioned together with someone from politician, musician, artist or athlete. Thus allows us to make leaps from one domain to another.

There are, however, several problems with the phonological representation of foreign named entities in Chinese. Due to the nature of Chinese script, NE transliterations can be realized very differently. The following is a summary of several problems that have to be taken into account:

- *word ending*: 阿拉法 vs.阿拉法特 ”Arafat” or 穆巴拉 vs.穆巴拉克 ”Mubarak”. The final consonant is not always transliterated. XIN transliterations tend to try to represent all phonemes and often add vowels to a final consonant to form a new syllable, whereas CNA transliteration tends to be shorter and may simply leave out a final consonant.

- *gender dependent choice of characters*: 萊絲 莉 ”Leslie” vs.萊斯利 ”Chris” or 克莉絲特 vs. 克莉斯

[2] To facilitate processing, the complete CGC was segmented and POS tagged using the Academia Sinica segmentation and tagging system (Ma and Huang, 2006).

[3] http://wordsketch.ling.sinica.edu.tw

特. Some occidental names are gender neutral. However, the choice of characters in a personal name in Chinese is often gender sensitive. So these names are likely to be transliterated differently depending on the gender of its referent.

- *divergent representations caused by scope of transliteration, e.g. both given and surname vs. only surname*: 大威廉絲 / 維・威廉絲 ”Venus Williams”.

- *difference in phonological interpretation*: 賴夫特 vs. 拉夫特 ”Rafter” or 康諾斯 vs. 康那斯 ”Connors”.

- *native vs. non-native pronunciation*: 艾斯庫 德 vs. 伊斯庫德 ”Escudero” or 費德洛 vs. 費德勒 ”Federer”.

2.1 Data collection

All data were collected from Chinese Gigaword Corpus using Chinese Sketch Engine with `WSDiff` function, which provides side-by-side syntagmatic comparison of Word Sketches for two different words. `WSDiff` query for w_i and w_j returns patterns that are common for both words and also patterns that are particular for each of them. Three data sets are thus provided. We neglect the common patterns set and concentrate only on the wordlists specific for each word.

2.2 Pairs extraction

Transliteration pairs are extracted from the two sets, A and B, collected with `WSDiff` using default set of seed pairs :

- for each seed pair in seeds retrieve `WSDiff` for `and/or` relation, thus have pairs of word lists, $< A_i, B_i >$
- for each word $w_{ii} \in A_i$ find best matching counterpart(s) $w_{ij} \in B_i$. Comparison is done using simple phonological rules, viz. 2.3
- use newly extracted pairs as new seeds (original seeds are stored as good pairs and not queried any more)
- loop until there are no new pairs

Notice that even though substantial proportion of borrowing among different communities, there is no mixing in the local context of collocation, which means, local collocation could be the most reliable way to detect language variants with known variants.

2.3 Phonological comparison

All word forms are converted from Chinese script into a phonological representation[4] during the pairs extraction phase and then these representations are compared and similarity scores are given to all pair candidates.

A lot of Chinese characters have multiple pronunciations and thus multiple representations are derived. In case of multiple pronunciations for certain syllable, this syllable is commpared to its counterpart from the other set. E.g. (葉 has three pronunciations: *yè, xié, shè*. When comparing syllables such as 裴[pei,fei] and 斐[fei], 裴 will be represented as [fei]. In case of pairs such as 葉爾欽 [ye er qin] and 葉爾侵 [ye er qin], which have syllables with multiple pronunciations and this multiple representations. However, since these two potential variants share the first two characters (out of three), they are considered as variants without superfluous phonological checking.

Phonological representations of whole words are then compared by Levenstein algorithm, which is widely used to measure the similarity between two strings. First, each syllable is split into initial and final components: *gao*:*g*+*ao*. In case of syllables without initials like *er*, an *'* is inserted before the syllable, thus *er*:*'*+*er*.

Before we ran the Levenstein measure, we also apply phonological corrections on each pair of candidate representations. Rules used for these corrections are derived from phonological features of Mandarin Chinese and extended with few rules from observation of the data: (1) For **Initials**, (a): voiced/voiceless stop contrasts are considered as similar for initials: *g:k*, e.g. 高 [gao] (高爾) vs. 科 [ke] (科爾),*d:t*, *b:p*, (b): *r:l* 瑞 [rui] (柯吉瑞夫) 列 [lie] (科濟列夫) is added to distinctive feature set based on observation. (2). For **Finals**, (a): pair *ei:ui* is evaluated as equivalent.[5] (b): oppositions of nasalised final is evaluated as dissimilar.

[4] http://unicode.org/charts/unihan.html

[5] *Pinyin* representation of phonology of Mandarin Chinese does not follow the phonological reality exactly: [ui] = [uei] etc.

2.4 Extraction algorithm

Our algorithm will potentially exhaust the whole corpus, i.e. find most of the named entities that occur with at least few other names entities, but only if seeds are chosen wisely and cover different domains[6]. However, some domains might not overlap at all, that is, members of those domains never appear in the corpus in relation `and/or`. And concurrence of members within some domains might be sparser than in other, e.g. politicians tend to be mentioned together more often than novelists. Nature of the corpus also plays important role. It is likely to retrieve more `and/or` related names from journalistic style. This is one of the reasons why we chose Chinese Gigaword Corpus for this task.

3 Experiment and evaluation

We have tested our method on the Chinese Gigaword Second Edition corpus with 11 manually selected seeds Apart from the selection of the starter seeds, the whole process is fully automatic. For this task we have collected data from syntagmatic relation `and/or`, which contains words co-occurring frequently with our seed words. When we make a query for peoples names, it is expected that most of the retrieved items will also be names, perhaps also names of locations, organizations etc.

The whole experiment took 505 iterations in which 494 pairs were extracted.

Our complete experiment with 11 pre-selected transliteration pairs as seed took 505 iterations to end. The iterations identified 494 effective transliteration variant pairs (i.e. those which were not among the seeds or pairs identified by earlier iteration.) All the 494 candidate pairs were manually evaluated 445 of them are found to be actual contrast pairs, a precision of 90.01%. In addition, the number of new transliteration pairs yielded is 4,045%, a very productive yield for NE discovery.

Preliminary results show that this approach is competitive against other approaches reported in previous studies. Performances of our algorithms is calculated in terms of precision rate with 90.01%.

4 Conclusion and Future work

In this paper, we have shown that it is possible to identify NE's without having prior knowledge of them. We also showed that, applying WSE to restrict grammatical context and saliency of collocation, we are able to effectively extract transliteration variants in a language where transliteration is not explicitly represented. We also show that a small set of seeds is all it needs for the proposed method to identify hundreds of transliteration variants. This proposed method has important applications in information retrieval and data mining in Chinese data.

In the future, we will be experimenting with a different set of seeds in a different domain to test the robustness of this approach, as well as to discover transliteration variants in our fields. We will also be focusing on more refined phonological analysis. In addition, we would like to explore the possibility of extending this proposal to other language pairs.

References

Jiang, L. and M.Zhou and L.f. Chien. 2007. *Named Entity Discovery based on Transliteration and WWW* [In Chinese]. Journal of the Chinese Information Processing Society. 2007 no.1. pp.23-29.

Graff, David et al. 2005. *Chinese Gigaword Second Edition.* Linguistic Data Consortium, Philadelphia.

Ma, Wei-Yun and Huang, Chu-Ren. 2006. *Uniform and Effective Tagging of a Heterogeneous Giga-word Corpus.* Presented at the 5th International Conference on Language Resources and Evaluation (LREC2006), 24-28 May. Genoa, Italy.

Kilgarriff, Adam et al. 2004. *The Sketch Engine.* Proceedings of EURALEX 2004. Lorient, France.

Paola Virga and Sanjeev Khudanpur. 2003. *Transliteration of proper names in cross-lingual information retrieval.* In Proc. of the ACL Workshop on Multilingual Named Entity Recognition, pp.57-64.

Wan, Stephen and Cornelia Verspoor. 1998. *Automatic English-Chinese Name Transliteration for Development of Multiple Resources.* In Proc. of COLING/ACL, pp.1352-1356.

[6] The term domain refers to *politics,music,sport, film* etc.

Detecting Semantic Relations between Named Entities in Text Using Contextual Features

Toru Hirano, Yoshihiro Matsuo, Genichiro Kikui
NTT Cyber Space Laboratories, NTT Corporation
1-1 Hikarinooka, Yokosuka-Shi, Kanagawa, 239-0847, Japan
{hirano.tohru, matsuo.yoshihiro, kikui.genichiro}@lab.ntt.co.jp

Abstract

This paper proposes a supervised learning method for detecting a semantic relation between a given pair of named entities, which may be located in different sentences. The method employs newly introduced contextual features based on centering theory as well as conventional syntactic and word-based features. These features are organized as a tree structure and are fed into a boosting-based classification algorithm. Experimental results show the proposed method outperformed prior methods, and increased precision and recall by 4.4% and 6.7%.

1 Introduction

Statistical and machine learning NLP techniques are now so advanced that named entity (NE) taggers are in practical use. Researchers are now focusing on extracting semantic relations between NEs, such as "George Bush (*person*)" is "president (*relation*)" of "the United States (*location*)", because they provide important information used in information retrieval, question answering, and summarization.

We represent a semantic relation between two NEs with a tuple [NE$_1$, NE$_2$, Relation Label]. Our final goal is to extract tuples from a text. For example, the tuple [George Bush (*person*), the U.S. (*location*), president (*Relation Label*)] would be extracted from the sentence "George Bush is the president of the U.S.". There are two tasks in extracting tuples from text. One is detecting whether or not a given pair of NEs are semantically related (*relation detection*), and the other is determining the relation label (*relation characterization*).

In this paper, we address the task of relation detection. So far, various supervised learning approaches have been explored in this field (Culotta and Sorensen, 2004; Zelenko et al., 2003). They use two kinds of features: syntactic ones and word-based ones, for example, the path of the given pair of NEs in the parse tree and the word n-gram between NEs (Kambhatla, 2004).

These methods have two problems which we consider in this paper. One is that they target only intra-sentential relation detection in which NE pairs are located in the same sentence, in spite of the fact that about 35% of NE pairs with semantic relations are inter-sentential (See Section 3.1). The other is that the methods can not detect semantic relations correctly when NE pairs located in a parallel sentence arise from a predication ellipsis. In the following Japanese example[1], the syntactic feature, which is the path of two NEs in the dependency structure, of the pair with a semantic relation ("Ken$_{11}$" and "Tokyo$_{12}$") is the same as the feature of the pair with no semantic relation ("Ken$_{11}$" and "New York$_{14}$").

(S-1) *Ken*$_{11}$*-wa Tokyo*$_{12}$*-de, Tom*$_{13}$*-wa*
New York$_{14}$*-de umareta*$_{15}$.
(Ken$_{11}$ was born$_{15}$ in Tokyo$_{12}$, Tom$_{13}$ in
New York$_{14}$.)

To solve the above problems, we propose a supervised learning method using contextual features.

The rest of this paper is organized as follows. Section 2 describes the proposed method. We report the results of our experiments in Section 3 and conclude the paper in Section 4.

2 Relation Detection

The proposed method employs contextual features based on centering theory (Grosz et al., 1983) as well as conventional syntactic and word-based features. These features are organized as a tree structure and are fed into a boosting-based classification algorithm. The method consists of three parts: preprocessing (POS tagging, NE tagging, and parsing),

[1] The numbers show correspondences of words between Japanese and English.

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 157–160,
Prague, June 2007. ©2007 Association for Computational Linguistics

feature extraction (contextual, syntactic, and word-based features), and classification.

In this section, we describe the underlying idea of contextual features and how contextual features are used for detecting semantic relations.

2.1 Contextual Features

When a pair of NEs with a semantic relation appears in different sentences, the antecedent NE must be contextually easily referred to in the sentence with the following NE. In the following Japanese example, the pair “Ken$_{22}$” and “amerika$_{32}$ (the U.S.)” have a semantic relation “wataru$_{33}$ (go)”, because “Ken$_{22}$” is contextually referred to in the sentence with “amerika$_{32}$” (In fact, the zero pronoun ϕ_i refers to “Ken$_{22}$”). Meanwhile, the pair “Naomi$_{25}$” and “amerika$_{32}$” has no semantic relation, because the sentence with “amerika$_{32}$” does not refer to “Naomi$_{25}$”.

(S-2) *asu*$_{21}$*, Ken*$_{22}$*-wa Osaka*$_{23}$*-o otozure*$_{24}$ *Naomi*$_{25}$*-to au*$_{26}$*.*
(Ken$_{22}$ is going to visit$_{24}$ Osaka$_{23}$ to see$_{26}$ Naomi$_{25}$, tomorrow$_{21}$.)

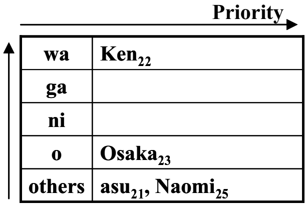
(S-3) *sonogo*$_{31}$*, (*ϕ_i*-ga) amerika*$_{32}$*-ni watari*$_{33}$ *Tom*$_{34}$*-to ryoko*$_{35}$ *suru.*
(Then$_{31}$, (he$_i$) will go$_{33}$ to the U.S.$_{32}$ to travel$_{35}$ with Tom$_{34}$.)

Furthermore, when a pair of NEs with a semantic relation appears in a parallel sentence arise from predication ellipsis, the antecedent NE is contextually easily referred to in the phrase with the following NE. In the example of “(S-1)”, the pair “Ken$_{11}$” and “Tokyo$_{12}$” have a semantic relation “umareta$_{15}$ (was born)”. Meanwhile, the pair “Ken$_{11}$” and “New York$_{14}$” has no semantic relation.

Therefore, using whether the antecedent NE is referred to in the context with the following NE as features of a given pair of NEs would improve relation detection performance. In this paper, we use centering theory (Kameyama, 1986) to determine how easily a noun phrase can be referred to in the following context.

2.2 Centering Theory

Centering theory is an empirical sorting rule used to identify the antecedents of (zero) pronouns. When there is a (zero) pronoun in the text, noun phrases that are in the previous context of the pronoun are sorted in order of likelihood of being the antecedent. The sorting algorithm has two steps. First, from the beginning of the text until the pronoun appears, noun phrases are stacked depending on case markers such as particles. In the above example, noun phrases, “asu$_{21}$”, “Ken$_{22}$”, “Osaka$_{23}$” and “Naomi$_{25}$”, which are in the previous context of the zero pronoun ϕ_i, are stacked and then the information shown in Figure 1 is acquired. Second, the stacked information is sorted by the following rules.



wa	Ken$_{22}$
ga	
ni	
o	Osaka$_{23}$
others	asu$_{21}$, Naomi$_{25}$

Figure 1: Information Stacked According to Centering Theory

1. The priority of case markers is as follows: “wa > ga > ni > o > others”
2. The priority of stack structure is as follows: last-in first-out, in the same case marker

For example, Figure 1 is sorted by the above rules and then the order, 1: “Ken$_{22}$”, 2: “Osaka$_{23}$”, 3: “Naomi$_{25}$”, 4: “asu$_{21}$”, is assigned. In this way, using centering theory would show that the antecedent of the zero pronoun ϕ_i is “Ken$_{22}$”.

2.3 Applying Centering Theory

When detecting a semantic relation between a given pair of NEs, we use centering theory to determine how easily the antecedent NE can be referred to in the context with the following NE. Note that we do not explicitly execute anaphora resolutions here.

Applied centering theory to relation detection is as follows. First, from the beginning of the text until the following NE appears, noun phrases are stacked depending on case markers, and the stacked information is sorted by the above rules (Section 2.2). Then, if the top noun phrase in the sorted order is identical to the antecedent NE, the antecedent NE is ”positive” when being referred to in the context with the following NE.

When the pair of NEs, “Ken$_{22}$” and “amerika$_{32}$”, is given in the above example, the noun phrases, “asu$_{21}$”, “Ken$_{22}$”, “Osaka$_{23}$” and “Naomi$_{25}$”, which are in the previous context of the following NE “amerika$_{32}$”, are stacked (Figure 1). Then they are sorted by the above sorting rules and the order, 1: “Ken$_{22}$”, 2: “Osaka$_{23}$”, 3: “Naomi$_{25}$”, 4: “asu$_{21}$”, is acquired. Here, because the top noun phrase in the sorted order is identical to the antecedent NE, the antecedent NE “Ken$_{22}$” is ”positive” when be-

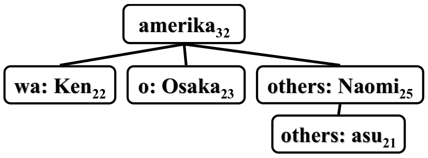


Figure 2: Centering Structure

ing referred to in the context with the following NE "amerika$_{32}$". Whether or not the antecedent NE is referred to in the context with the following NE is used as a feature. We call this feature Centering Top (CT).

2.4 Using Stack Structure

The sorting algorithm using centering theory tends to rank highly thoes words that easily become subjects. However, for relation detection, it is necessary to consider both NEs that easily become subjects, such as person and organization, and NEs that do not easily become subjects, such as location and time.

We use the stack described in Section 2.3 as a structural feature for relation detection. We call this feature Centering Structure (CS). For example, the stacked information shown in Figure 1 is assumed to be structure information, as shown in Figure 2. The method of converting from a stack (Figure 1) into a structure (Figure 2) is described as follows. First, the following NE, "amerika$_{32}$", becomes the root node because Figure 1 is stacked information until the following NE appears. Then, the stacked information is converted to Figure 2 depending on the case markers. We use the path of the given pair of NEs in the structure as a feature. For example, "amerika$_{32}$ → wa:Ken$_{22}$"[2] is used as the feature of the given pair "Ken$_{22}$" and "amerika$_{32}$".

2.5 Classification Algorithm

There are several structure-based learning algorithms proposed so far (Collins and Duffy, 2001; Suzuki et al., 2003; Kudo and Matsumoto, 2004). The experiments tested Kudo and Matsumoto's boosting-based algorithm using sub trees as features, which is implemented as the BACT system.

In relation detection, given a set of training examples each of which represents contextual, syntactic, and word-based features of a pair of NEs as a tree labeled as either having semantic relations or not, the BACT system learns that a set of rules are effective in classifying. Then, given a test instance, which represents contextual, syntactic, and word-based features of a pair of NEs as a tree, the BACT system classifies using a set of learned rules.

Type	% of pairs with semantic relations
(A) Intra-sentential	31.4% (3333 / 10626)
(B) Inter-sentential	0.8% (1777 / 225516)
(A)+(B) Total	2.2% (5110 / 236142)

Table 1: Percent of pairs with semantic relations in annotated text

3 Experiments

We experimented with texts from Japanese newspapers and weblogs to test the proposed method. The following four models were compared:

1. **WD** : Pairs of NEs within n words are detected as pairs with semantic relation.
2. **STR** : Supervised learning method using syntactic[3] and word-based features, the path of the pairs of NEs in the parse tree and the word n-gram between pairs of NEs (Kambhatla, 2004)
3. **STR-CT** : STR with the centering top feature explained in Section 2.3.
4. **STR-CS** : STR with the centering structure feature explained in Section 2.4.

3.1 Setting

We used 1451 texts from Japanese newspapers and weblogs, whose semantic relations between person and location had been annotated by humans for the experiments[4]. There were 5110 pairs with semantic relations out of 236,142 pairs in the annotated text. We conducted ten-fold cross-validation over 236,142 pairs of NEs so that sets of pairs from a single text were not divided into the training and test sets.

We also divided pairs of NEs into two types: (A) intra-sentential and (B) inter-sentential. The reason for dividing them is so that syntactic structure features would be effective in type (A) and contextual features would be effective in type (B). Another reason is that the percentage of pairs with semantic relations out of the total pairs in the annotated text differ significantly between types, as shown in Table 1.

In the experiments, all features were automatically acquired using a Japanese morphological and dependency structure analyzer.

[2] "A → B" means A has a dependency relation to B.

[3] There is no syntactic feature in inter-sentential.

[4] We are planning to evaluate the other pairs of NEs.

	(A)+(B) Total		(A) Intra-sentential		(B) Inter-sentential	
	Precision	Recall	Precision	Recall	Precsion	Recall
WD10	43.0(2501/5819)	48.9(2501/5110)	48.1(2441/5075)	73.2(2441/3333)	8.0(60/744)	3.4(60/1777)
STR	69.3(2562/3696)	50.1(2562/5110)	75.6(2374/3141)	71.2(2374/3333)	33.9(188/555)	10.6(188/1777)
STR-CT	71.4(2764/3870)	54.1(2764/5110)	78.4(2519/3212)	75.6(2519/3333)	37.2(245/658)	13.8(245/1777)
STR-CS	73.7(2902/3935)	56.8(2902/5110)	80.1(2554/3187)	76.6(2554/3333)	46.5(348/748)	27.6(348/1777)

WD10: NE pairs that appear within 10 words are detected.

Table 2: Results for Relation Detection

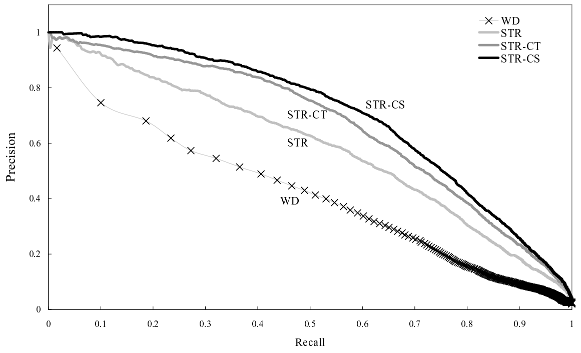


Figure 3: Recall-precision Curves: (A)+(B) total

3.2 Results

To improve relation detection performance, we investigated the effect of the proposed method using contextual features. Table 2 shows results for Type (A), Type (B), and (A)+(B). We also plotted recall-precision curves[5], altering threshold parameters, as shown in Figure 3.

The comparison between STR and STR-CT and between STR and STR-CS in Figure 3 indicates that the proposed method effectively contributed to relation detection. In addition, the results for Type (A): intra-sentential, and (B): inter-sentential, in Table 2 indicate that the proposed method contributed to both Type (A), improving precision by about 4.5% and recall by about 5.4% and Type (B), improving precision by about 12.6% and recall by about 17.0%.

3.3 Error Analysis

Over 70% of the errors are covered by two major problems left in relation detection.

Parallel sentence: The proposed method solves problems, which result from when a parallel sentence arises from predication ellipsis. However, there are several types of parallel sentence that differ from the one we explained. (For example, Ken and Tom was born in Osaka and New York, respectively.)

Definite anaphora: Definite noun phrase, such as “Shusho (the Prime Minister)” and “Shacho (the President)”, can be anaphors. We should consider them in centering theory, but it is difficult to find them in Japanese .

4 Conclusion

In this paper, we propose a supervised learning method using words, syntactic structures, and contextual features based on centering theory, to improve both inter-sentential and inter-sentential relation detection. The experiments demonstrated that the proposed method increased precision by 4.4%, up to 73.7%, and increased recall by 6.7%, up to 56.8%, and thus contributed to relation detection.

In future work, we plan to solve the problems relating to parallel sentence and definite anaphora, and address the task of relation characterization.

References

M. Collins and N. Duffy. 2001. Convolution Kernels for Natural Language. *Proceedings of the Neural Information Processing Systems*, pages 625–632.

A. Culotta and J. Sorensen. 2004. Dependency Tree Kernels for Relation Extraction. *Annual Meeting of Association of Computational Linguistics*, pages 423–429.

B. J. Grosz, A. K. Joshi, and S. Weistein. 1983. Providing a unified account of definite nounphrases in discourse. *Annual Meeting of Association of Computational Linguistics*, pages 44–50.

N. Kambhatla. 2004. Combining Lexical, Syntactic, and Semantic Features with Maximum Entropy Models for Information Extraction. *Annual Meeting of Association of Computational Linguistics*, pages 178–181.

M. Kameyama. 1986. A property-sharing constraint in centering. *Annual Meeting of Association of Computational Linguistics*, pages 200–206.

T. Kudo and Y. Matsumoto. 2004. A boosting algorithm for classification of semi-structured text. *In Proceedings of the 2004 EMNLP*, pages 301–308.

J. Suzuki, T. Hirao, Y. Sasaki, and E. Maeda. 2003. Hierarchical directed acyclic graph kernel : Methods for structured natural language data. *Annual Meeting of Association of Computational Linguistics*, pages 32–39.

D. Zelenko, C. Aone, and A. Richardella. 2003. Kernel Methods for Relation Extraction. *Journal of Machine Learning Research*, pages 3:1083–1106.

[5] Precision = # of correctly detected pairs / # of detected pairs Recall = # of correctly detected pairs / # of pairs with semantic relations

Mapping Concrete Entities from PAROLE-SIMPLE-CLIPS to ItalWordNet: Methodology and Results

Adriana Roventini, Nilda Ruimy, Rita Marinelli, Marisa Ulivieri, Michele Mammini
Istituto di Linguistica Computazionale – CNR
Via Moruzzi,1 – 56124 – Pisa, Italy
{adriana.roventini,nilda.ruimy,rita.marinelli,
marisa.ulivieri,michele.mammini}@ilc.cnr.it

Abstract

This paper describes a work in progress aiming at linking the two largest Italian lexical-semantic databases ItalWordNet and PAROLE-SIMPLE-CLIPS. The adopted linking methodology, the software tool devised and implemented for this purpose and the results of the first mapping phase regarding 1stOrderEntities are illustrated here.

1 Introduction

The mapping and the integration of lexical resources is today a main concern in the world of computational linguistics. In fact, during the past years, many linguistic resources were built whose bulk of linguistic information is often neither easily accessible nor entirely available, whereas their visibility and interoperability would be crucial for HLT applications.

The resources here considered constitute the largest and extensively encoded Italian lexical semantic databases. Both were built at the CNR Institute of Computational Linguistics, in Pisa.

The ItalWordNet lexical database (henceforth IWN) was first developed in the framework of EuroWordNet project and then enlarged and improved in the national project SI-TAL[1]. The theoretical model underlying this lexicon is based on the EuroWordNet lexical model (Vossen, 1998) which is, in its turn, inspired to the Princeton WordNet (Fellbaum, 1998).

PAROLE-SIMPLE-CLIPS (PSC) is a four-level lexicon developed over three different projects: the LE-PAROLE project for the morphological and syntactic layers, the LE-SIMPLE project for the semantic model and lexicon and the Italian project CLIPS[2] for the phonological level and the extension of the lexical coverage. The theoretical model underlying this lexicon is based on the EAGLES recommendations, on the results of the EWN and ACQUILEX projects and on a revised version of Pustejovsky's Generative Lexicon theory (Pustejovsky 1995).

In spite of the different underlying principles and peculiarities characterizing the two lexical models, IWN and PSC lexicons also present many compatible aspects and the reciprocal enhancements that the linking of the resources would entail were illustrated in Roventini et al., (2002); Ruimy & Roventini (2005). This has prompted us to envisage the semi-automatic link of the two lexical databases, eventually merging the whole information into a common representation framework. The first step has been the mapping of the 1stOrderEntities which is described in the following.

This paper is organized as follows: in section 2 the respective ontologies and their mapping are briefly illustrated, in section 3 the methodology followed to link these resources is described; in section 4 the software tool and its workings are explained; section 5 reports on the results of the complete mapping of the 1stOrderEntities. Future work is outlined in the conclusion.

2 Mapping Ontology-based Lexical Resources

In both lexicons, the backbone for lexical representation is provided by an ontology of semantic types.

[1] *Integrated System for the Automatic Language Treatment.*

[2] *Corpora e Lessici dell'Italiano Parlato e Scritto.*

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 161–164,
Prague, June 2007. ©2007 Association for Computational Linguistics

The IWN Top Ontology (TO) (Roventini et al., 2003), which slightly differs from the EWN TO[3], consists in a hierarchical structure of 65 language-independent Top Concepts (henceforth TCs) clustered in three categories distinguishing 1st OrderEntities, 2ndOrderEntities and 3rdOrder Entities. Their subclasses, hierarchically ordered by means of a subsumption relation, are also structured in terms of (disjunctive and non-disjunctive) opposition relations. The IWN database is organized around the notion of *synset*, i.e. a set of synonyms. Each synset is ontologically classified on the basis of its hyperonym and connected to other synsets by means of a rich set of lexical-semantic relations. Synsets are in most cases cross-classified in terms of multiple, non disjoint TCs, e.g.: *informatica* (computer science): [Agentive, Purpose, Social, Unboundedevent]. The semantics of a word sense or *synset variant* is fully defined by its membership in a synset.

The SIMPLE Ontology (SO)[4], which consists of 157 language-independent semantic types, is a multidimensional type system based on hierarchical and non-hierarchical conceptual relations. In the type system, multidimensionality is captured by *qualia roles* that define the distinctive properties of semantic types and differentiate their internal semantic constituency. The SO distinguishes therefore between *simple* (one-dimensional) and *unified* (multi-dimensional) semantic types, the latter implementing the principle of *orthogonal inheritance.* In the PSC lexicon, the basic unit is the word sense, represented by a 'semantic unit' (henceforth, *SemU*). Each SemU is assigned one single semantic type (e.g.: *informatica*: [Domain]), which endows it with a structured set of semantic information.

A primary phase in the process of mapping two ontology-based lexical resources clearly consisted in establishing correspondences between the conceptual classes of both ontologies, with a view to further matching their respective instances.

The mapping will only be briefly outlined here for the 1stOrderEntity. More information can be found in (Ruimy & Roventini 2005; Ruimy, 2006).

The IWN 1stOrderEntity class structures concrete entities (referred to by concrete nouns). Its main cross-classifying subclasses: Form, Origin, Composition and Function correspond to the four Qualia roles the SIMPLE model avails of to express orthogonal aspects of word meaning. Their respective subdivisions consist of (mainly) disjoint classes, e.g. Natural vs. Artifact. To each class corresponds, in most of the cases, a SIMPLE semantic type or a type hierarchy subsumed by the Concrete_entity top type. Some other IWN TCs, such as Comestible, Liquid, are instead mappable to SIMPLE distinctive features: e.g. Plus_Edible, Plus_Liquid, etc.

3 Linking Methodology

Mapping is performed on a semantic type-driven basis. A semantic type of the SIMPLE ontology is taken as starting point. Considering the type's SemUs along with their PoS and 'isa' relation, the IWN resource is explored in search of linking candidates with same PoS and whose ontological classification matches the correspondences established between the classes of both ontologies.

A characteristic of this linking is that it involves lexical elements having a different status, i.e. semantic units and synsets.

During the linking process, two different types of data are returned from each mapping run:
1) A set of matched pairs of word senses, i.e. SemUs and synset variants with identical string, PoS and whose respective ontological classification perfectly matches. After human validation, these matched word senses are linked.
2) A set of unmatched word senses, in spite of their identical string and PoS value. Matching failure is due to a mismatch of the ontological classification of word senses existing in both resources. Such mismatch may be originated by:
a) an incomplete ontological information. As already explained, IWN synsets are cross-classified in terms of a combination of TCs; however, cases of synsets lacking some meaning component are not rare. The problem of incomplete ontological classification may often be overcome by relaxing the mapping constraints; yet, this solution can only be applied if the existing ontological label is informative enough. Far more problematic to deal with are those cases of incomplete or little informative ontological labels, e.g. 1stOrderEntities as different as *medicinale, anello, vetrata* (medicine, ring, picture window) and only classified as 'Function';

[3] A few changes were in fact necessary to allow the encoding of new syntactic categories.
[4] http://www.ilc.cnr.it/clips/Ontology.htm

b) a different ontological information. Besides mere encoding errors, ontological classification discrepancy may be imputable to:

i) a different but equally defensible meaning interpretation (e.g.: *ala* (aircraft wing) : [Part] vs. [Artifact Instrument Object]). Word senses falling into this category are clustered into numerically significant sets according to their semantic typing and then studied with a view to establishing further equivalences between ontological classes or to identify, in their classification schemes, descriptive elements lending themselves to be mapped.

ii) a different level of specificity in the ontological classification, due either to the lexicographer's subjectivity or to an objective difference of granularity of the ontologies.

The problems in ii) may be bypassed by climbing up the ontological hierarchy, identifying the parent nodes and allowing them to be taken into account in the mapping process.

Hyperonyms of matching candidates are taken into account during the linking process and play a particularly determinant role in the resolution of cases whereby matching fails due to a conflict of ontological classification. It is the case for sets of word senses displaying a different ontological classification but sharing the same hyperonym, e.g. *collana, braccialetto* (necklace, bracelet) typed as [Clothing] in PSC and as [Artifact Function] in IWN but sharing the hyperonym *gioiello* (jewel). Hyperonyms are also crucial for polysemous senses belonging to different semantic types in PSC but sharing the same ontological classification in IWN, e.g.: SemU1595*viola* (violet) [Plant] and SemU1596*viola* (violet) [Flower] vs. IWN: *viola*1 (has_hyperonym *pianta*1 (plant)) and *viola*3 (has_hyperonym *fiore*1 (flower)), both typed as [Group Plant].

4 The Linking Tool

The LINKPSC_IWN software tool implemented to map the lexical units of both lexicons works in a semiautomatic way using the ontological classifications, the 'isa' relations and some semantic features of the two resources. Since the 157 semantic types of the SO provide a more fine-grained structure of the lexicon than the 65 top concepts of the IWN ontology, which reflect only fundamental distinctions, mapping is PSC → IWN oriented. The mapping process foresees the following steps:

1) Selection of a PSC semantic type and definition of the loading criteria, i.e. either all its SemUs or only those bearing a given information;

2) Selection of one or more mapping constraints on the basis of the correspondences established between the conceptual classes of both ontologies, in order to narrow the automatic mapping;

3) Human validation of the automatic mapping and storage of the results;

4) If necessary, relaxation/tuning of the mapping constraints and new processing of the input data.

By human validation of the automatic mapping we also intend the manual selection of the semantically relevant word sense pair(s) from the set of possible matches automatically output for each SemU. A decision is taken after checking relevant information sources such as hyperonyms, SemU/synset glosses and the IWN-ILI link.

Besides the mapping results, a list of unmatched word senses is provided which contains possible encoding errors and polysemous senses of the considered SemUs (e.g., *kiwi* (fruit) which is discarded when mapping the 'Animal' class). Some of these word senses proceed from an extension of meaning, e.g. People-Human: *pigmeo, troglodita* (pygmy, troglodyte) or Animal-Human *verme, leone* (worm, lion) and are used with different levels of intentionality: either as a semantic surplus or as dead metaphors (Marinelli, 2006).

More interestingly, the list of unmatched words also contains the IWN word senses whose synset's ontological classification is incomplete or different w.r.t. the constraints imposed to the mapping run. Analyzing these data is therefore crucial to identify further mapping constraints. A list of PSC lexical units missing in IWN is also generated, which is important to appropriately assess the lexical intersection between the two resources.

5 Results

From a quantitative point of view three main issues are worth noting (cf. Table 1): first, the considerable percentage of linked senses with respect to the linkable ones (i.e. words with identical string and PoS value); second, the many cases of multiple mappings; third, the extent of overlapping coverage.

SemUs selected	27768	
Linkable senses	15193	54,71%
Linked senses	10988	72,32%
Multiple mappings	1125	10,23%
Unmatched senses	4205	27,67%

Table 1 summarizing data

Multiple mappings depend on the more fine grained sense distinctions performed in IWN. The eventual merging of the two resources would make up for such discrepancy.

During the linking process, many other possibilities of reciprocal improvement and enrichment were noticed by analyzing the lists of unmatched word-senses. All the inconsistencies are in fact recorded together with their differences in ontological classification, or in the polysemy treatment that the mapping evidenced. Some mapping failures have been observed due to a different approach to the treatment of polysemy in the two resources: for example, a single entry in PSC corresponding to two different IWN entries encoding very fined-grained nuances of sense, e.g. *galeotto*1 (galley rower) and *galeotto*2 (galley slave).

Other mapping failures are due to cases of encoding inconsistency. For example, when a word sense from a multi-variant synset is linked to a SemU, all the other variants from the same synset should map to PSC entries sharing the same semantic type, yet in some cases it has been observed that SemUs corresponding to variants of the same synset do not share a common semantic type.

All these encoding differences or inconsistencies were usefully put in the foreground by the linking process and are worthy of further in-depth analysis with a view to the merging, harmonization and interoperability of the two lexical resources.

6 Conclusion and Future Work

In this paper the PSC-IWN linking of concrete entities, the methodology adopted, the tool implemented to this aim and the results obtained are described. On the basis of the encouraging results illustrated here, the linking process will be carried on by dealing with 3rdOrder Entities. Our attention will then be devoted to 2ndOrderEntities which, so far, have only been object of preliminary investigations on Speech act (Roventini 2006) and Feeling verbs. Because of their intrinsic complexity, the linking of 2ndOrderEntities is expected to be a far more challenging task.

References

James Pustejovsky 1995. *The generative lexicon*. MIT Press.

Christiane Fellbaum (ed.) 1998. *Wordnet: An Electronic Lexical Database*. MIT Press.

Piek Vossen (ed.) 1998. EuroWordNet: *A multilingual database with lexical semantic networks*. Kluwer Academic Publishers.

Adriana Roventini et al. 2003. ItalWordNet: *Building a Large Semantic Database for the Automatic Treatment of Italian*. Computational Linguistics in Pisa, Special Issue, XVIII-XIX, Pisa-Roma, IEPI. Tomo II, 745--791.

Nilda Ruimy et al. 2003. *A computational semantic lexicon of Italian: SIMPLE*. In A. Zampolli, N. Calzolari, L. Cignoni, (eds.), Computational Linguistics in Pisa, Special Issue, XVIII-XIX, (2003). Pisa-Roma, IEPI. Tomo II, 821-864.

Adriana Roventini, Marisa Ulivieri and Nicoletta Calzolari. 2002 *Integrating two semantic lexicons, SIMPLE and ItalWordNet: what can we gain?* LREC Proceedings, Vol. V, pp. 1473-1477.

Nilda Ruimy and Adriana Roventini. 2005 *Towards the linking of two electronic lexical databases of Italian*, In Zygmunt Veutulani (ed.), L&T'05 -

Nilda Ruimy. 2006. *Merging two Ontology-based Lexical Resources*. LREC Proceedings, CD-ROM, 1716-1721.

Adriana Roventini. 2006. *Linking Verbal Entries of Different Lexical Resources*. LREC Proceedings, CD-ROM, 1710-1715.

Rita Marinelli. 2006. *Computational Resources and Electronic Corpora in Metaphors Evaluation*. Second International Conference of the German Cognitive Linguistics Association, Munich, 5-7 October.

Extracting Hypernym Pairs from the Web

Erik Tjong Kim Sang
ISLA, Informatics Institute
University of Amsterdam
erikt@science.uva.nl

Abstract

We apply pattern-based methods for collecting hypernym relations from the web. We compare our approach with hypernym extraction from morphological clues and from large text corpora. We show that the abundance of available data on the web enables obtaining good results with relatively unsophisticated techniques.

1 Introduction

WordNet is a key lexical resource for natural language applications. However its coverage (currently 155k synsets for the English WordNet 2.0) is far from complete. For languages other than English, the available WordNets are considerably smaller, like for Dutch with a 44k synset WordNet. Here, the lack of coverage creates bigger problems. A manual extension of the WordNets is costly. Currently, there is a lot of interest in automatic techniques for updating and extending taxonomies like WordNet.

Hearst (1992) was the first to apply fixed syntactic patterns like *such NP as NP* for extracting hypernym-hyponym pairs. Carballo (1999) built noun hierarchies from evidence collected from conjunctions. Pantel, Ravichandran and Hovy (2004) learned syntactic patterns for identifying hypernym relations and combined these with clusters built from co-occurrence information. Recently, Snow, Jurafsky and Ng (2005) generated tens of thousands of hypernym patterns and combined these with noun clusters to generate high-precision suggestions for unknown noun insertion into WordNet (Snow et al., 2006). The previously mentioned papers deal with English. Little work has been done for other languages. IJzereef (2004) used fixed patterns to extract Dutch hypernyms from text and encyclopedias. Van der Plas and Bouma (2005) employed noun distribution characteristics for extending the Dutch part of EuroWordNet.

In earlier work, different techniques have been applied to large and very large text corpora. Today, the web contains more data than the largest available text corpus. For this reason, we are interested in employing the web for the extraction of hypernym relations. We are especially curious about whether the size of the web allows to achieve meaningful results with basic extraction techniques.

In section two we introduce the task, hypernym extraction. Section three presents the results of our web extraction work as well as a comparison with similar work with large text corpora. Section four concludes the paper.

2 Task and Approach

We examine techniques for extending WordNets. In this section we describe the relation we focus on, introduce our evaluation approach and explain the query format used for obtaining web results.

2.1 Task

We concentrate on a particular semantic relation: hypernymy. One term is a hypernym of another if its meaning both covers the meaning of the second term and is broader. For example, *furniture* is a hypernym of *table*. The opposite term for hypernym is hyponym. So *table* is a hyponym of *furniture*. Hypernymy is a transitive relation. If term A is a hypernym of term B while term B is a hypernym of term

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 165–168,
Prague, June 2007. ©2007 Association for Computational Linguistics

C then term A is also a hypernym of term C.

In WordNets, hypernym relations are defined between senses of words (synsets). The Dutch WordNet (Vossen, 1998) contains 659,284 of such hypernym noun pairs of which 100,268 are immediate links and 559,016 are inherited by transitivity. More importantly, the resource contains hypernym information for 45,979 different nouns. A test with a Dutch newspaper text revealed that the WordNet only covered about two-thirds of the noun lemmas in the newspaper (among the missing words were *e-mail*, *euro* and *provider*). Proper names pose an even larger problem: the Dutch WordNet only contains 1608 words that start with a capital character.

2.2 Collecting evidence

In order to find evidence for the existence of hypernym relations between words, we search the web for fixed patterns like *H such as A, B and C*. Following Snow et al. (2006), we derive two types of evidence from these patterns:

- *H* is a hypernym of *A*, *B* and *C*
- *A*, *B* and *C* are siblings of each other

Here, *sibling* refers to the relative position of the words in the hypernymy tree. Two words are siblings of each other if they share a parent.

We compute a hypernym evidence score $S(h, w)$ for each candidate hypernym h for word w. It is the sum of the normalized evidence for the hypernymy relation between h and w, and the evidence for sibling relations between w and known hyponyms s of h:

$$S(h, w) = \frac{f_{hw}}{\sum_x f_{xw}} + \sum_s \frac{g_{sw}}{\sum_y g_{yw}}$$

where f_{hw} is the frequency of patterns that predict that h is a hypernym of w, g_{sw} is the frequency of patterns that predict that s is a sibling of w, and x and y are arbitrary words from the WordNet. For each word w, we select the candidate hypernym h with the largest score $S(h, w)$.

For each hyponym, we only consider evidence for hypernyms and siblings. We have experimented with different scoring schemes, for example by including evidence from hypernyms of hypernyms and remote siblings, but found this basic scoring scheme to perform best.

2.3 Evaluation

We use the Dutch part of EuroWordNet (DWN) (Vossen, 1998) for evaluation of our hypernym extraction methods. Hypernym-hyponym pairs that are present in the lexicon are assumed to be correct. In order to have access to negative examples, we make the same assumption as Snow et al. (2005): the hypernymy relations in the WordNets are complete for the terms that they contain. This means that if two words are present in the lexicon without the target relation being specified between them, then we assume that this relation does not hold between them. The presence of positive and negative examples allows for an automatic evaluation in which precision, recall and F values are computed.

We do not require our search method to find the exact position of a target word in the hypernymy tree. Instead, we are satisfied with any ancestor. In order to rule out identification methods which simply return the top node of the hierarchy for all words, we also measure the distance between the assigned hypernym and the target word. The ideal distance is one, which would occur if the suggested ancestor is a parent. Grandparents are associated with distance two and so on.

2.4 Composing web queries

In order to collect evidence for lexical relations, we search the web for lexical patterns. When working with a fixed corpus on disk, an exhaustive search can be performed. For web search, however, this is not possible. Instead, we rely on acquiring interesting lexical patterns from text snippets returned for specific queries. The format of the queries has been based on three considerations.

First, a general query like *such as* is insufficient for obtaining much interesting information. Most web search engines impose a limit on the number of results returned from a query (for example 1000), which limits the opportunities for assessing the performance of such a general pattern. In order to obtain useful information, the query needs to be more specific. For the pattern *such as*, we have two options: adding the hypernym, which gives *hypernym such as*, or adding the hyponym, which results in *such as hyponym*.

Both extensions of the general pattern have their limitations. A pattern that includes the hypernym may fail to generate enough useful information if the hypernym has many hyponyms. And patterns with hyponyms require more queries than patterns with hypernyms (one per child rather than one per parent). We chose to include hyponyms in the patterns. This approach models the real world task in which one is looking for the meaning of an unknown entity.

The final consideration regards which hyponyms to use in the queries. Our focus is on evaluating the approach via comparison with an existing WordNet. Rather than submitting queries for all 45,979 nouns in the lexical resource to the web search engine, we will use a random sample of nouns.

3 Hypernym extraction

We describe our web extraction work and compare the results with our earlier work with extraction from a text corpus and hypernym prediction from morphological information.

3.1 Earlier work

In earlier work (Tjong Kim Sang and Hofmann, 2007), we have applied different methods for obtaining hypernym candidates for words. First, we extracted hypernyms from a large text corpus (300Mwords) following the approach of Snow et al. (2006). We collected 16728 different contexts in which hypernym-hyponym pairs were found and evaluated individual context patterns as well as a combination which made use of Bayesian Logistic Regression. We also examined a single pattern predicting only sibling relations: A *en*(*and*) B.

Additionally, we have applied a corpus-independent morphological approach which takes advantage of the fact that in Dutch, compound words often have the head in the final position (like *blackbird* in English). The head is a good hypernym candidate for the compound and therefore long words which end with a legal Dutch word often have this suffix as hypernym (Sabou et al., 2005).

The results of the approaches can be found in Table 1. The corpus approaches achieve reasonable precision rates. The recall scores are low because we attempt to retrieve a hypernym for *all* nouns in the WordNet. Surprisingly enough the basic morphological approach outperforms all corpus methods, both with respect to precision and recall.

Method	**Prec.**	**Recall**	**F**	**Dist.**
corpus: *N zoals N*	0.22	0.0068	0.013	2.01
corpus: combined	0.36	0.020	0.038	2.86
corpus: *N en N*	0.31	0.14	0.19	1.98
morphological approach	0.54	0.33	0.41	1.19

Table 1: Performances measured in our earlier work (Tjong Kim Sang and Hofmann, 2007) with a morphological approach and patterns applied to a text corpus (single hypernym pattern, combined hypernym patterns and single conjunctive pattern). Predicting valid suffixes of words as their hypernyms, outperforms the corpus approaches.

3.2 Extraction from the web

For our web extraction work, we used the same individual extraction patterns as in the corpus work: *zoals* (*such as*) and *en* (*and*), but not the combined hypernym patterns because the expected performance did not make up for the time complexity involved. We added randomly selected candidate hyponyms to the queries to improve the chance to retrieve interesting information.

This approach worked well. As Table 2 shows, for both patterns the recall score improved in comparison with the corpus experiments. Additionally, the single web hypernym pattern *zoals* outperformed the combination of corpus hypernym patterns with respect to recall and distance. Again, the conjunctive pattern outperformed the hypernym pattern. We assume that the frequency of the two patterns plays an important role (the frequency of pages with the conjunctive pattern is five times the frequency of pages with *zoals*).

Finally, we combined word-internal information with the conjunctive pattern approach by adding the morphological candidates to the web evidence before computing hypernym pair scores. This approach achieved the highest recall score at only slight precision loss (Table 2).

3.3 Error analysis

We have inspected the output of the conjunctive web extraction with word-internal information. For this purpose we have selected the ten most frequent hypernym pairs (Table 3), the ten least frequent and the ten pairs exactly between these two groups. 40%

Method	**Prec.**	**Recall**	**F**	**Dist.**
web: *N zoals N*	0.23	0.089	0.13	2.06
web: *N en N*	0.39	0.31	0.35	2.04
morphological approach	0.54	0.33	0.41	1.19
web: *en* + morphology	0.48	0.45	0.46	1.64

Table 2: Performances measured in the two web experiments and a combination of the best web approach with the morphological approach. The conjunctive web pattern *N en N* rates best, because of its high frequency. The recall rate can be improved by supplying the best web approach with word-internal information.

of the pairs were correct, 47% incorrect and 13% were plausible but contained relations that were not present in the reference WordNet. In the center group of ten pairs all errors are caused by the morphological approach while all other errors originate from the web extraction method.

4 Concluding remarks

The contributions of this paper are two-fold. First, we show that the large quantity of available web data allows basic patterns to perform better on hypernym extraction than a combination of extraction patterns applied to a large corpus. Second, we demonstrate that the performance of web extraction can be improved by combining its results with those of a corpus-independent morphological approach.

The described approach is already being applied in a project for extending the coverage of the Dutch WordNet. However, we remain interested in obtaining a better performance levels especially in higher recall scores. There are some suggestions on how we could achieve this. First, our present selection method, which ignores all but the first hypernym suggestion, is quite strict. We expect that the lower-ranked hypernyms include a reasonable number of correct candidates as well. Second, a combination of web patterns most likely outperforms individual patterns. Obtaining results for many different web pattens will be a challenge given the restrictions on the number of web queries we can currently use.

+/-	**score**	**hyponym**	**hypernym**
-	912	buffel	predator
+	762	trui	kledingstuk
?	715	motorfiets	motorrijtuig
+	697	kruidnagel	specerij
-	680	concours	samenzijn
+	676	koopwoning	woongelegenheid
+	672	inspecteur	opziener
?	660	roller	werktuig
?	654	rente	verdiensten
?	650	cluster	afd.

Table 3: Example output of the the conjunctive web system with word-internal information. Of the ten most frequent pairs, four are correct (+). Four others are plausible but are missing in the WordNet (?).

References

Sharon A. Caraballo. 1999. Automatic construction of a hypernym-labeled noun hierarchy from text. In *Proceedings of ACL-99*. Maryland, USA.

Marti A. Hearst. 1992. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of ACL-92*. Newark, Delaware, USA.

Leonie IJzereef. 2004. *Automatische extractie van hyperniemrelaties uit grote tekstcorpora*. MSc thesis, University of Groningen.

Patrick Pantel, Deepak Ravichandran, and Eduard Hovy. 2004. Towards terascale knowledge acquisition. In *Proceedings of COLING 2004*, pages 771–777. Geneva, Switzerland.

Lonneke van der Plas and Gosse Bouma. 2005. Automatic acquisition of lexico-semantic knowledge for qa. In *Proceedings of the IJCNLP Workshop on Ontologies and Lexical Resources*. Jeju Island, Korea.

Marta Sabou, Chris Wroe, Carole Goble, and Gilad Mishne. 2005. Learning domain ontologies for web service descriptions: an experiment in bioinformatics. In *14th International World Wide Web Conference (WWW2005)*. Chiba, Japan.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2005. Learning syntactic patterns for automatic hypernym discovery. In *NIPS 2005*. Vancouver, Canada.

Rion Snow, Daniel Jurafsky, and Andrew Y. Ng. 2006. Semantic taxonomy induction from heterogenous evidence. In *Proceedings of COLING/ACL 2006*. Sydney, Australia.

Erik Tjong Kim Sang and Katja Hofmann. 2007. Automatic extraction of dutch hypernym-hyponym pairs. In *Proceedings of the Seventeenth Computational Linguistics in the Netherlands*. Katholieke Universiteit Leuven, Belgium.

Piek Vossen. 1998. *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Kluwer Academic Publisher.

An OWL Ontology for HPSG

Graham Wilcock
University of Helsinki
PO Box 9
00014 Helsinki, Finland
graham.wilcock@helsinki.fi

Abstract

The paper presents an OWL ontology for HPSG. The HPSG ontology is integrated with an existing OWL ontology, GOLD, as a community of practice extension. The basic ideas are illustrated by visualizations of type hierarchies for parts of speech.

1 Introduction

The paper presents an OWL ontology for HPSG (Head-driven Phrase Structure Grammar) (Sag et al., 2003). OWL is the W3C Web Ontology Language (http://www.w3.org/2004/OWL). An existing ontology is used as a starting point: GOLD (Section 2) is a general ontology for linguistic description. As HPSG is a more specific linguistic theory, the HPSG ontology (Section 3) is integrated inside GOLD as a sub-ontology known as a community of practice extension (Section 4).

2 GOLD: A General Ontology for Linguistic Description

GOLD, a General Ontology for Linguistic Description (http://www.linguistics-ontology.org/) (Farrar and Langendoen, 2003) is an OWL ontology that aims to capture “the general knowledge of the field that is usually possessed by a well trained linguist. This includes knowledge that potentially forms the basis of any theoretical framework. In particular, GOLD captures the fundamentals of descriptive linguistics. Examples of such knowledge are ‘a verb is a part of speech’, ‘gender can be semantically grounded’, or ‘linguistic expressions realize morphemes’.” (Farrar and Lewis, 2005).

As far as possible GOLD uses language-neutral and theory-neutral terminology. For instance, parts of speech are subclasses of *gold:GrammaticalUnit* as shown in Figure 1. As GOLD is language-neutral, a wide range of parts of speech are included. For example, both Preposition and Postposition are included as subclasses of Adposition. The classes in the OWLViz graphical visualization (on the right in Figure 1) have been selected from the complete list in the Asserted Hierarchy (on the left).

Originally GOLD was intended to be neutral where linguistic theories had divergent views, but a recent development is the idea of supporting different sub-communities as communities of practice (Farrar and Lewis, 2005) within the GOLD framework. A community of practice may focus on developing a consensus in a specific area, for example in phonology or in Bantu languages. On the other hand, communities of practice may focus on competing theories, where each sub-community has its own distinctive terminology and divergent conceptualization. In this case, the aim is to capture explicitly the relationship between the sub-community view and the overall framework, in the form of a Community Of Practice Extension (COPE) (Farrar and Lewis, 2005). A COPE is a sub-ontology that inherits from, and extends, the overall GOLD ontology. Sub-ontology classes are distinguished from each other by different namespace prefixes, for example *gold:Noun* and *hpsg:noun*.

3 An OWL Ontology for HPSG

HPSG OWL is an OWL ontology for HPSG that is currently under development. As the aims of the first version of the ontology are clarity and acceptability,

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 169–172,
Prague, June 2007. ©2007 Association for Computational Linguistics

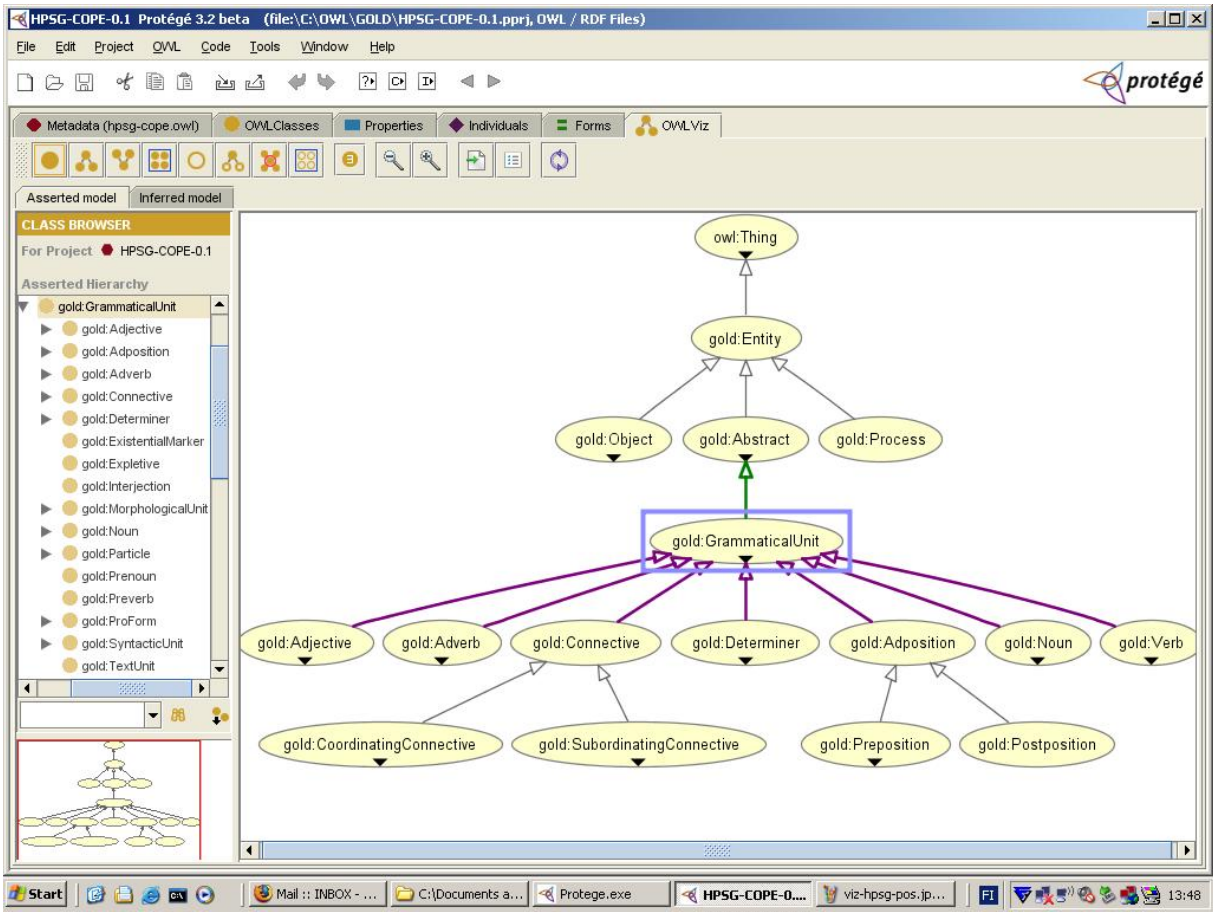


Figure 1: Parts of speech in GOLD

it carefully follows the standard textbook version of HPSG by Sag et al. (2003). This also means that the first version is English-specific, as the core grammars presented in the textbook are English-specific.

In HPSG OWL, parts of speech are subclasses of *hpsg:pos*, as shown in Figure 2. As this version is English-specific, it has prepositions (*hpsg:prep*) but not postpositions. Parts of speech that have agreement features (in English) form a distinct subclass *hpsg:agr-pos* including *hpsg:det* (determiner) and *hpsg:verb*. Within *hpsg:agr-pos*, *hpsg:comp* (complementizer) and *hpsg:noun* form a further subclass *hpsg:nominal*. This particular conceptualization of the type hierarchy is specific to (Sag et al., 2003).

The Protégé-OWL (http://protege.stanford.edu) ontology editor supports both visual construction and visual editing of the hierarchy. For example, if *hpsg:adj* had agreement features, it could be moved under *hpsg:agr-pos* by a simple drag-and-drop (in the Asserted Hierarchy pane on the left). Both the visualization (in the OWLViz pane on the right) and the underlying OWL statements (not shown) are automatically generated. The grammar writer does not edit OWL statements directly.

This is a significant advantage of the new technology over current grammar development tools. For example, LKB (Copestake, 2002) can produce a visualization of the type hierarchy from the underlying Type Definition Language (TDL) statements, but the hierarchy can only be modified by textually editing the TDL statements.

4 A Community of Practice Extension

HPSG COPE is a community of practice extension that integrates the HPSG ontology within GOLD. The COPE is an OWL ontology that imports both the GOLD and the HPSG ontologies. Apart from the import statements, the COPE consists entirely of

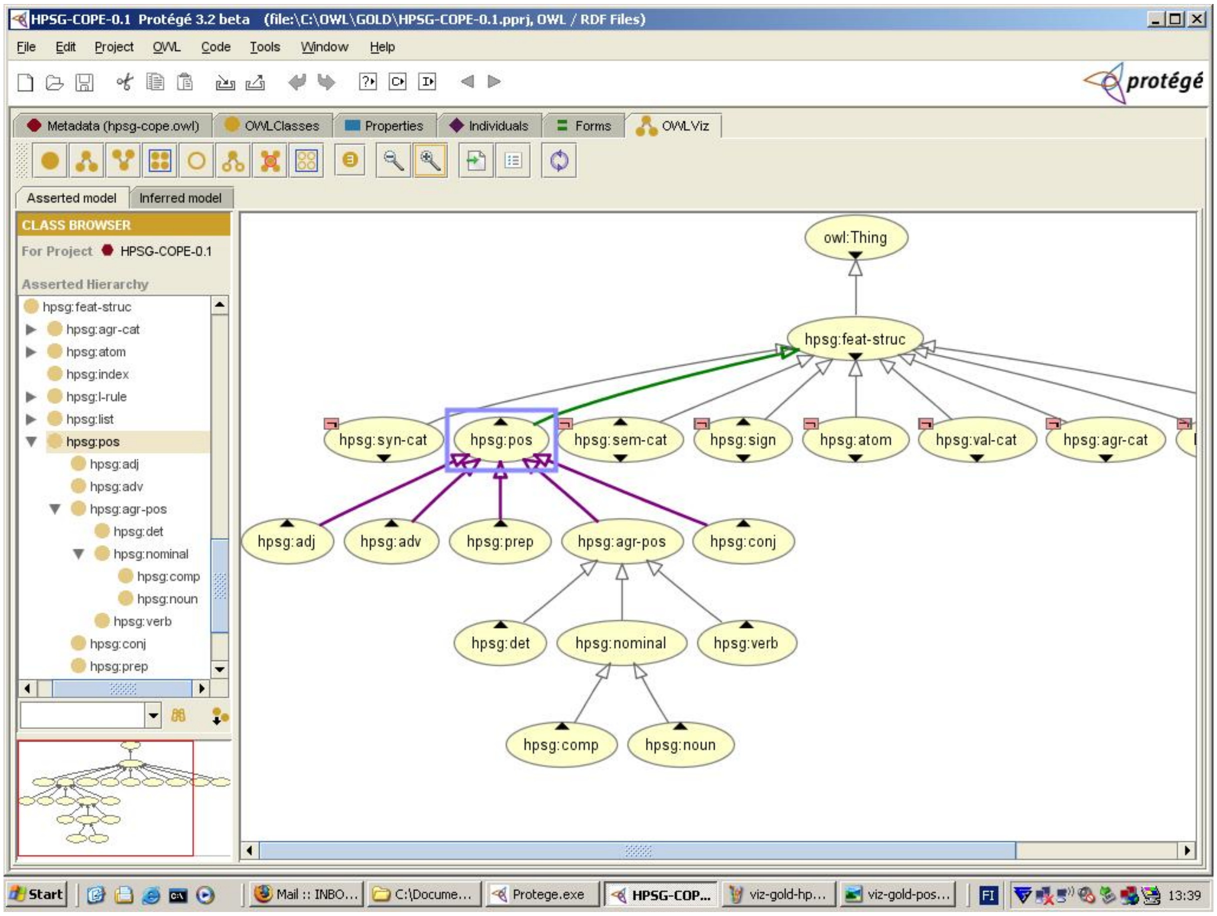


Figure 2: Parts of speech in HPSG

rdfs:subClassOf and *rdfs:subPropertyOf* statements. HPSG COPE defines HPSG classes as subclasses of GOLD classes and HPSG properties as subproperties of GOLD properties.

In the COPE, parts of speech in HPSG are subsumed by appropriate parts of speech in GOLD, as shown in Figure 3. In some cases this is straightforward, for example *hpsg:adj* is mapped to *gold:Adjective*. In other cases, the HPSG theory-specific terminology differs significantly from the theory-neutral terminology in GOLD. Some of the mappings are based on definitions of the HPSG terms given in a glossary in (Sag et al., 2003), for example the mapping of *hpsg:conj* (conjunction) to *gold:CoordinatingConnective* and the mapping of *hpsg:comp* (complementizer) to *gold:SubordinatingConnective*.

Properties in HPSG OWL are defined by HPSG COPE as subproperties of GOLD properties. For example, the HPSG OWL class *hpsg:sign* (Sag et al., 2003) (p. 475) properties:

PHON	type: list (a sequence of word forms)
SYN	type: gram-cat (a grammatical category)
SEM	type: sem-struc (a semantic structure)

are mapped to the GOLD class *gold:LinguisticSign* properties:

hasForm	Range: PhonologicalUnit
hasGrammar	Range: GrammaticalUnit
hasMeaning	Range: SemanticUnit

by the HPSG COPE *rdfs:subPropertyOf* definitions:

hpsg:PHON	subproperty of	gold:hasForm
hpsg:SYN	subproperty of	gold:hasGrammar
hpsg:SEM	subproperty of	gold:hasMeaning

5 Conclusion

The paper has described an initial version of an OWL ontology for HPSG, together with an approach to integrating it with GOLD as a community of prac-

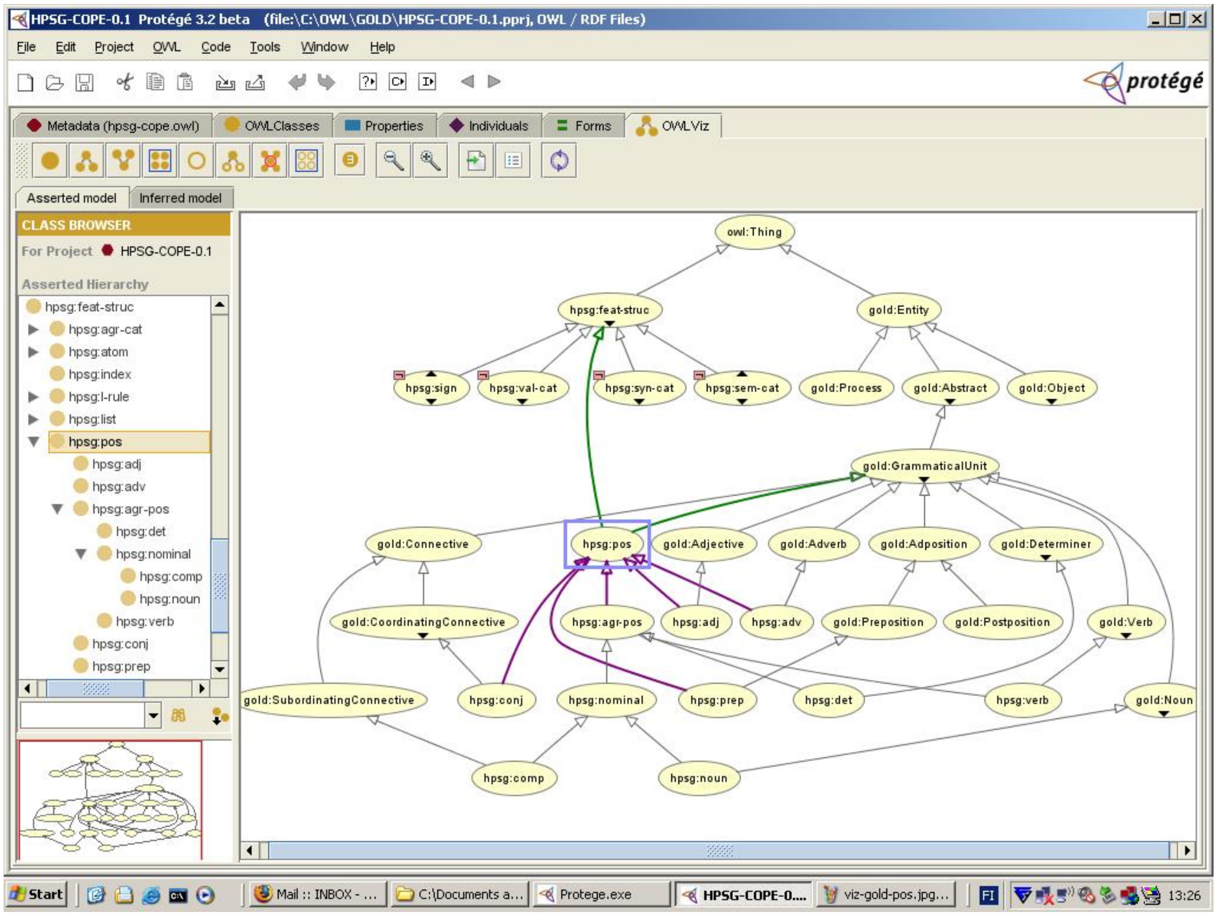


Figure 3: Parts of speech in the Community of Practice Extension

tice extension. Perhaps a rigorous foundation of typed feature structures and a clear type hierarchy makes HPSG more amenable to expression as an ontology than other linguistic theories.

Protégé-OWL supports visual development and visual editing of the ontology. This is a significant practical advantage over existing grammar development tools. OWLViz provides graphical visualizations of any part of the ontology.

OWL DL (Description Logic) reasoners can be run inside Protégé to check consistency and to do cross-classification. One current research topic is how to exploit reasoners to perform automatically the kind of cross-classification that is widely used in HPSG linguistic analyses.

Another current topic is how to implement HPSG lexical rules and grammar rules in the ontology. An interesting possibility is to use the W3C Semantic Web Rule Language, SWRL (Wilcock, 2006).

References

Ann Copestake. 2002. *Implementing Typed Feature Structure Grammars*. CSLI Publications, Stanford, CA.

Scott Farrar and D. Terence Langendoen. 2003. A linguistic ontology for the semantic web. *GLOT International*, 7.3:97–100.

Scott Farrar and William D. Lewis. 2005. The GOLD Community of Practice: An infrastructure for linguistic data on the web. http://www.u.arizona.edu/˜farrar/.

Ivan A. Sag, Thomas Wasow, and Emily Bender. 2003. *Syntactic Theory: A Formal Introduction*. CSLI Publications, Stanford, CA.

Graham Wilcock. 2006. Natural language parsing with GOLD and SWRL. In *RuleML-2006, Rules and Rule Markup Languages for the Semantic Web (Online Proceedings)*, Athens, GA.

Classifying Temporal Relations Between Events

Nathanael Chambers and **Shan Wang** and **Dan Jurafsky**
Department of Computer Science
Stanford University
Stanford, CA 94305
{natec,shanwang,jurafsky}@stanford.edu

Abstract

This paper describes a fully automatic two-stage machine learning architecture that learns temporal relations between pairs of events. The first stage learns the temporal attributes of single event descriptions, such as tense, grammatical aspect, and aspectual class. These imperfect guesses, combined with other linguistic features, are then used in a second stage to classify the temporal relationship between two events. We present both an analysis of our new features and results on the TimeBank Corpus that is 3% higher than previous work that used perfect human tagged features.

1 Introduction

Temporal information encoded in textual descriptions of events has been of interest since the early days of natural language processing. Lately, it has seen renewed interest as Question Answering, Information Extraction and Summarization domains find it critical in order to proceed beyond surface understanding. With the recent creation of the Timebank Corpus (Pustejovsky et al., 2003), the utility of machine learning techniques can now be tested.

Recent work with the Timebank Corpus has revealed that the six-class classification of temporal relations is very difficult, even for human annotators. The highest score reported on Timebank achieved 62.5% accuracy when using gold-standard features as marked by humans (Mani et al., 2006). This paper describes an approach using features extracted automatically from raw text that not only duplicates this performance, but surpasses its accuracy by 3%. We do so through advanced linguistic features and a surprising finding that using automatic rather than hand-labeled tense and aspect knowledge causes only a slight performance degradation.

We briefly describe current work on temporal ordering in section 2. Section 4 describes the first stage of basic temporal extraction, followed by a full description of the second stage in 5. The evaluation and results on Timebank then follow in section 6.

2 Previous Work

Mani et. al (2006) built a MaxEnt classifier that assigns each pair of events one of 6 relations from an augmented Timebank corpus. Their classifier relies on perfect features that were hand-tagged in the corpus, including tense, aspect, modality, polarity and event class. Pairwise agreement on tense and aspect are also included. In a second study, they applied rules of temporal transitivity to greatly expand the corpus, providing different results on this enlarged dataset. We could not duplicate their reported performance on this enlarged data, and instead focus on performing well on the Timebank data itself.

Lapata and Lascarides (2006) trained an event classifier for inter-sentential events. They built a corpus by saving sentences that contained two events, one of which is triggered by a key time word (e.g. *after* and *before*). Their learner was based on syntax and clausal ordering features. Boguraev and Ando (2005) evaluated machine learning on related tasks, but not relevant to event-event classification.

Our work is most similar to Mani's in that we are

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 173–176,
Prague, June 2007.

learning relations given event pairs, but our work extends their results both with new features and by using fully automatic linguistic features from raw text that are not hand selected from a corpus.

3 Data

We used the Timebank Corpus (v1.1) for evaluation, 186 newswire documents with 3345 event pairs. Solely for comparison with Mani, we add the 73 document Opinion Corpus (Mani et al., 2006) to create a larger dataset called the OTC. We present both Timebank and OTC results so future work can compare against either. All results below are from 10-fold cross validation.

4 Stage One: Learning Event Attributes

The task in Stage One is to learn the five temporal attributes associated with events as tagged in the Timebank Corpus. (1) *Tense* and (2) grammatical *aspect* are necessary in any approach to temporal ordering as they define both temporal location and structure of the event. (3) *Modality* and (4) *polarity* indicate hypothetical or non-occuring situations, and finally, (5) *event class* is the type of event (e.g. process, state, etc.). The *event class* has 7 values in Timebank, but we believe this paper's approach is compatible with other class divisions as well. The range of values for each event attribute is as follows, also found in (Pustejovsky et al., 2003):

tense	none, present, past, future
aspect	none, prog, perfect, prog_perfect
class	report, aspectual, state, I_state I_action, perception, occurrence
modality	none, to, should, would, could can, might
polarity	positive, negative

4.1 Machine Learning Classification

We used a machine learning approach to learn each of the five event attributes. We implemented both Naive Bayes and Maximum Entropy classifiers, but found Naive Bayes to perform as well or better than Maximum Entropy. The results in this paper are from Naive Bayes with Laplace smoothing.

The features we used on this stage include part of speech tags (two before the event), lemmas of the event words, WordNet synsets, and the appearance of auxiliaries and modals before the event. This latter set included all derivations of *be* and *have* auxiliaries, modal words (e.g. may, might, etc.), and the presence/absence of *not*. We performed feature selection on this list of features, learning a different set of features for each of the five attributes. The list of selected features for each is shown in figure 1.

tense	POS-2-event, POS-1-event POS-of-event, have_word, be_word
aspect	POS-of-event, modal_word, be_word
class	synset
modality	none
polarity	none

Figure 1: Features selected for learning each temporal attribute. POS-2 is two tokens before the event.

Timebank Corpus			
	tense	aspect	class
Baseline	52.21	84.34	54.21
Accuracy	88.28	94.24	75.2
Baseline (OTC)	48.52	86.68	59.39
Accuracy (OTC)	87.46	88.15	76.1

Figure 2: Stage One results on classification.

Modality and *polarity* did not select any features because their majority class baselines were so high (98%) that learning these attributes does not provide much utility. A deeper analysis of event interaction would require a modal analysis, but it seems that a newswire domain does not provide great variation in modalities. Consequently, modality and polarity are not used in Stage Two. Tense, aspect and class are shown in figure 2 with majority class baselines. Tense classification achieves 36% absolute improvement, aspect 10% and class 21%. Performance on the OTC set is similar, although aspect is not as good. These guesses are then passed to Stage Two.

5 Stage Two: Event-Event Features

The task in this stage is to choose the temporal relation between two events, given the pair of events. We assume that the events have been extracted and that there exists some relation between them; the task is to choose the relation. The Timebank Corpus uses relations that are based on Allen's set of thirteen (Allen, 1984). Six of the relations are inverses of the other six, and so we condense the set to *before*, *ibefore*, *includes*, *begins*, *ends* and *simultaneous*. We map the thirteenth *identity* into *simultaneous*. One oddity is that Timebank includes both *during* and *included_by* relations, but *during* does not appear in Timebank documentation. While we don't know how previous work handles this, we condense *during* into *included_by* (invert to *includes*).

5.1 Features

Event Specific: The five temporal attributes from Stage One are used for each event in the pair, as well as the event strings, lemmas and WordNet synsets. Mani added two other features from these, indicators if the events agree on tense and aspect. We add a third, event class agreement. Further, to capture the dependency between events in a discourse, we create new bigram features of tense, aspect and class (e.g. "present past" if the first event is in the present, and the second past).

Part of Speech: For each event, we include the Penn Treebank POS tag of the event, the tags for the two tokens preceding, and one token following. We use the Stanford Parser[1] to extract them. We also extend previous work and create bigram POS features of the event and the token before it, as well as the bigram POS of the first event and the second event.

Event-Event Syntactic Properties: A phrase P is said to dominate another phrase Q if Q is a daughter node of P in the syntactic parse tree. We leverage the syntactic output of the parser to create the *dominance* feature for intra-sentential events. It is either on or off, depending on the two events' syntactic dominance. Lapata used a similar feature for subordinate phrases and an indicator *before* for textual event ordering. We adopt these features and also add a *same-sentence* indicator if the events appear in the same sentence.

Prepositional Phrase: Since preposition heads are often indicators of temporal class, we created a new feature indicating when an event is part of a prepositional phrase. The feature's values range over 34 English prepositions. Combined with event dominance (above), these two features capture direct intra-sentential relationships. To our knowledge, we are the first to use this feature in temporal ordering.

Temporal Discourse: Seeing tense as a type of anaphora, it is a natural conclusion that the relationship between two events becomes stronger as the textual distance draws closer. Because of this, we adopted the view that intra-sentential events are generated from a different distribution than inter-sentential events. We therefore train two models during learning, one for events in the same sentence, and the other for events crossing sentence boundaries. It essentially splits the data on the *same_sentence* feature. As we will see, this turned out to be a very useful feature. It is called the *split* approach in the next section.

Example (require, compromise):
*"Their solution **required** a **compromise**..."*
Features
(lemma1: require) (lemma2: compromise) (dominates: yes) (tense-bigram: past-none) (aspect-bigram: none-none) (tense-match: no) (aspect-match: yes) (before: yes) (same-sent: yes)

6 Evaluation and Results

All results are from a 10-fold cross validation using SVM (Chang and Lin, 2001). We also evaluated Naive Bayes and Maximum Entropy. Naive Bayes (NB) returned similar results to SVM and we present feature selection results from NB to compare the added value of our new features.

The input to Stage Two is a list of pairs of events; the task is to classify each according to one of six temporal relations. Four sets of results are shown in figure 3. *Mani*, *Mani+Lapata* and *All+New* correspond to performance on features as listed in the figure. The three table columns indicate how a gold-standard Stage One (*Gold*) compares against imperfect guesses (*Auto*) and the guesses with split distributions (*Auto-Split*).

A clear improvement is seen in each row, indicating that our new features provide significant improvement over previous work. A decrease in performance is seen between columns *gold* and *auto*, as expected, because imperfect data is introduced, however, the drop is manageable. The *auto-split* distributions make significant gains for the Mani and Lapata features, but less when all new features are

[1] http://nlp.stanford.edu/software/lex-parser.shtml

Timebank Corpus	**Gold**	**Auto**	**Auto-Split**
Baseline	37.22	37.22	46.58
Mani	50.97	50.19	53.42
Mani+Lapata	52.29	51.57	55.10
All+New	60.45	59.13	**59.43**

Mani stage one attributes, tense/aspect-match, event strings
Lapata dominance, before, lemma, synset
New prep-phrases, same-sent, class-match, POS uni/bigrams, tense/aspect/class-bigrams

Figure 3: Incremental accuracy by adding features.

Same Sentence		**Diff Sentence**	
POS-1 Ev1	2.5%	Tense Pair	1.6%
POS Bigram Ev1	3.5%	Aspect Ev1	0.5%
Preposition Ev1	2.0%	POS Bigram	0.2%
Tense Ev2	0.7%	POS-1 Ev2	0.3%
Preposition Ev2	0.6%	Word EV2	0.2%

Figure 4: Top 5 features as added in feature selection w/ Naive Bayes, with their percentage improvement.

involved. The highest fully-automatic accuracy on Timebank is 59.43%, a 4.3% gain from our new features. We also report 67.57% *gold* and 65.48% *auto-split* on the OTC dataset to compare against Mani's reported hand-tagged features of 62.5%, a gain of 3% with our automatic features.

7 Discussion

Previous work on OTC achieved classification accuracy of 62.5%, but this result was based on "perfect data" from human annotators. A low number from good data is at first disappointing, however, we show that performance can be improved through more linguistic features and by isolating the distinct tasks of ordering inter-sentential and intra-sentential events.

Our new features show a clear improvement over previous work. The features that capture dependencies between the events, rather than isolated features provide the greatest utility. Also, the impact of imperfect temporal data is surprisingly minimal. Using Stage One's results instead of gold values hurts performance by less than 1.4%. This suggests that much of the value of the hand-coded information can be achieved via automatic approaches. Stage One's *event class* shows room for improvement, yet the negative impact on Event-Event relationships is manageable. It is conceivable that more advanced features would better classify the *event class*, but improvement on the event-event task would be slight.

Finally, it is important to note the difference in classifying events in the same sentence vs. cross-boundary. Splitting the 3345 pairs of corpus events into two separate training sets makes our data more sparse, but we still see a performance improvement when using Mani/Lapata features. Figure 4 gives a hint to the difference in distributions as the best features of each task are very different. Intra-sentence events rely on syntax cues (e.g. preposition phrases and POS), while inter-sentence events use tense and aspect. However, the differences are minimized as more advanced features are added. The final row in figure 3 shows minimal split improvement.

8 Conclusion

We have described a two-stage machine learning approach to event-event temporal relation classification. We have shown that imperfect event attributes can be used effectively, that a range of event-event dependency features provide added utility to a classifier, and that events within the same sentence have distinct characteristics from those across sentence boundaries. This fully automatic raw text approach achieves a 3% improvement over previous work based on perfect human tagged features.

Acknowledgement: This work was supported in part by the DARPA GALE Program and the DTO AQUAINT Program.

References

James Allen. 1984. Towards a general theory of action and time. *Artificial Intelligence*, 23:123–154.

Branimir Boguraev and Rie Kubota Ando. 2005. Timeml-compliant text analysis for temporal reasoning. In *IJCA-05*.

Chih-Chung Chang and Chih-Jen Lin, 2001. *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

Mirella Lapata and Alex Lascarides. 2006. Learning sentence-internal temporal relations. In *Journal of AI Research*, volume 27, pages 85–117.

Inderjeet Mani, Marc Verhagen, Ben Wellner, Chong Min Lee, and James Pustejovsky. 2006. Machine learning of temporal relations. In *ACL-06*, July.

James Pustejovsky, Patrick Hanks, Roser Sauri, Andrew See, David Day, Lisa Ferro, Robert Gaizauskas, Marcia Lazo, Andrea Setzer, and Beth Sundheim. 2003. The timebank corpus. *Corpus Linguistics*, pages 647–656.

Moses: Open Source Toolkit for Statistical Machine Translation

Philipp Koehn
Hieu Hoang
Alexandra Birch
Chris Callison-Burch
University of Edinburgh[1]

Marcello Federico
Nicola Bertoldi
ITC-irst[2]

Brooke Cowan
Wade Shen
Christine Moran
MIT[3]

Richard Zens
RWTH Aachen[4]

Chris Dyer
University of Maryland[5]

Ondřej Bojar
Charles University[6]

Alexandra Constantin
Williams College[7]

Evan Herbst
Cornell[8]

[1] pkoehn@inf.ed.ac.uk, {h.hoang, A.C.Birch-Mayne}@sms.ed.ac.uk, callison-burch@ed.ac.uk. [2] {federico, bertoldi}@itc.it. [3] brooke@csail.mit.edu, swade@ll.mit.edu, weezer@mit.edu. [4] zens@i6.informatik.rwth-aachen.de. [5] redpony@umd.edu. [6] bojar@ufal.ms.mff.cuni.cz. [7] 07aec_2@williams.edu. [8] evh4@cornell.edu

Abstract

We describe an open-source toolkit for statistical machine translation whose novel contributions are (a) support for linguistically motivated factors, (b) confusion network decoding, and (c) efficient data formats for translation models and language models. In addition to the SMT decoder, the toolkit also includes a wide variety of tools for training, tuning and applying the system to many translation tasks.

1 Motivation

Phrase-based statistical machine translation (Koehn et al. 2003) has emerged as the dominant paradigm in machine translation research. However, until now, most work in this field has been carried out on proprietary and in-house research systems. This lack of openness has created a high barrier to entry for researchers as many of the components required have had to be duplicated. This has also hindered effective comparisons of the different elements of the systems.

By providing a free and complete toolkit, we hope that this will stimulate the development of the field. For this system to be adopted by the community, it must demonstrate performance that is comparable to the best available systems. Moses has shown that it achieves results comparable to the most competitive and widely used statistical machine translation systems in translation quality and run-time (Shen et al. 2006). It features all the capabilities of the closed sourced Pharaoh decoder (Koehn 2004).

Apart from providing an open-source toolkit for SMT, a further motivation for Moses is to extend phrase-based translation with factors and confusion network decoding.

The current phrase-based approach to statistical machine translation is limited to the mapping of small text chunks without any explicit use of linguistic information, be it morphological, syntactic, or semantic. These additional sources of information have been shown to be valuable when integrated into pre-processing or post-processing steps.

Moses also integrates confusion network decoding, which allows the translation of ambiguous input. This enables, for instance, the tighter integration of speech recognition and machine translation. Instead of passing along the one-best output of the recognizer, a network of different word choices may be examined by the machine translation system.

Efficient data structures in Moses for the memory-intensive translation model and language model allow the exploitation of much larger data resources with limited hardware.

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 177–180,
Prague, June 2007. ©2007 Association for Computational Linguistics

2 Toolkit

The toolkit is a complete out-of-the-box translation system for academic research. It consists of all the components needed to preprocess data, train the language models and the translation models. It also contains tools for tuning these models using minimum error rate training (Och 2003) and evaluating the resulting translations using the BLEU score (Papineni et al. 2002).

Moses uses standard external tools for some of the tasks to avoid duplication, such as GIZA++ (Och and Ney 2003) for word alignments and SRILM for language modeling. Also, since these tasks are often CPU intensive, the toolkit has been designed to work with Sun Grid Engine parallel environment to increase throughput.

In order to unify the experimental stages, a utility has been developed to run repeatable experiments. This uses the tools contained in Moses and requires minimal changes to set up and customize.

The toolkit has been hosted and developed under sourceforge.net since inception. Moses has an active research community and has reached over 1000 downloads as of 1st March 2007.

The main online presence is at

http://www.statmt.org/moses/

where many sources of information about the project can be found. Moses was the subject of this year's Johns Hopkins University Workshop on Machine Translation (Koehn et al. 2006).

The decoder is the core component of Moses. To minimize the learning curve for many researchers, the decoder was developed as a drop-in replacement for Pharaoh, the popular phrase-based decoder.

In order for the toolkit to be adopted by the community, and to make it easy for others to contribute to the project, we kept to the following principles when developing the decoder:

- Accessibility
- Easy to Maintain
- Flexibility
- Easy for distributed team development
- Portability

It was developed in C++ for efficiency and followed modular, object-oriented design.

3 Factored Translation Model

Non-factored SMT typically deals only with the surface form of words and has one phrase table, as shown in Figure 1.

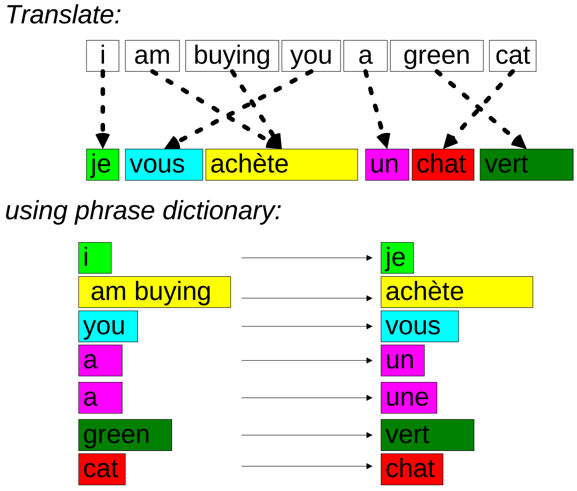


Figure 1. Non-factored translation

In factored translation models, the surface forms may be augmented with different factors, such as POS tags or lemma. This creates a factored representation of each word, Figure 2.

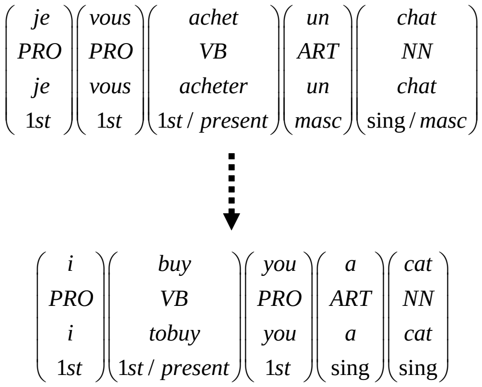


Figure 2. Factored translation

Mapping of source phrases to target phrases may be decomposed into several steps. Decomposition of the decoding process into various steps means that different factors can be modeled separately. Modeling factors in isolation allows for flexibility in their application. It can also increase accuracy and reduce sparsity by minimizing the number dependencies for each step.

For example, we can decompose translating from surface forms to surface forms and lemma, as shown in Figure 3.

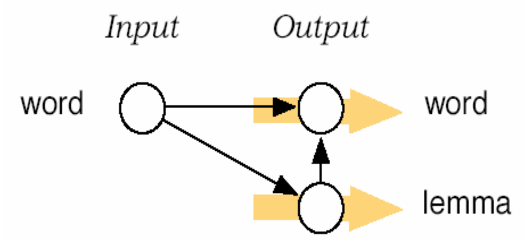


Figure 3. Example of graph of decoding steps

By allowing the graph to be user definable, we can experiment to find the optimum configuration for a given language pair and available data.

The factors on the source sentence are considered fixed, therefore, there is no decoding step which create source factors from other source factors. However, Moses can have ambiguous input in the form of confusion networks. This input type has been used successfully for speech to text translation (Shen et al. 2006).

Every factor on the target language can have its own language model. Since many factors, like lemmas and POS tags, are less sparse than surface forms, it is possible to create a higher order language models for these factors. This may encourage more syntactically correct output. In Figure 3 we apply two language models, indicated by the shaded arrows, one over the words and another over the lemmas. Moses is also able to integrate factored language models, such as those described in (Bilmes and Kirchhoff 2003) and (Axelrod 2006).

4 Confusion Network Decoding

Machine translation input currently takes the form of simple sequences of words. However, there are increasing demands to integrate machine translation technology into larger information processing systems with upstream NLP/speech processing tools (such as named entity recognizers, speech recognizers, morphological analyzers, etc.). These upstream processes tend to generate multiple, erroneous hypotheses with varying confidence. Current MT systems are designed to process only one input hypothesis, making them vulnerable to errors in the input.

In experiments with confusion networks, we have focused so far on the speech translation case, where the input is generated by a speech recognizer. Namely, our goal is to improve performance of spoken language translation by better integrating speech recognition and machine translation models. Translation from speech input is considered more difficult than translation from text for several reasons. Spoken language has many styles and genres, such as, formal read speech, unplanned speeches, interviews, spontaneous conversations; it produces less controlled language, presenting more relaxed syntax and spontaneous speech phenomena. Finally, translation of spoken language is prone to speech recognition errors, which can possibly corrupt the syntax and the meaning of the input.

There is also empirical evidence that better translations can be obtained from transcriptions of the speech recognizer which resulted in lower scores. This suggests that improvements can be achieved by applying machine translation on a large set of transcription hypotheses generated by the speech recognizers and by combining scores of acoustic models, language models, and translation models.

Recently, approaches have been proposed for improving translation quality through the processing of multiple input hypotheses. We have implemented in Moses confusion network decoding as discussed in (Bertoldi and Federico 2005), and developed a simpler translation model and a more efficient implementation of the search algorithm. Remarkably, the confusion network decoder resulted in an extension of the standard text decoder.

5 Efficient Data Structures for Translation Model and Language Models

With the availability of ever-increasing amounts of training data, it has become a challenge for machine translation systems to cope with the resulting strain on computational resources. Instead of simply buying larger machines with, say, 12 GB of main memory, the implementation of more efficient data structures in Moses makes it possible to exploit larger data resources with limited hardware infrastructure.

A phrase translation table easily takes up gigabytes of disk space, but for the translation of a single sentence only a tiny fraction of this table is needed. Moses implements an efficient representation of the phrase translation table. Its key properties are a *prefix tree* structure for source words and *on demand loading*, i.e. only the fraction of the phrase table that is needed to translate a sentence is loaded into the working memory of the decoder.

For the Chinese-English NIST task, the memory requirement of the phrase table is reduced from 1.7 gigabytes to less than 20 mega bytes, with no loss in translation quality and speed (Zens and Ney 2007).

The other large data resource for statistical machine translation is the language model. Almost unlimited text resources can be collected from the Internet and used as training data for language modeling. This results in language models that are too large to easily fit into memory.

The Moses system implements a data structure for language models that is more efficient than the canonical SRILM (Stolcke 2002) implementation used in most systems. The language model on disk is also converted into this binary format, resulting in a minimal loading time during start-up of the decoder.

An even more compact representation of the language model is the result of the *quantization* of the word prediction and back-off probabilities of the language model. Instead of representing these probabilities with 4 byte or 8 byte floats, they are sorted into bins, resulting in (typically) 256 bins which can be referenced with a single 1 byte index. This quantized language model, albeit being less accurate, has only minimal impact on translation performance (Federico and Bertoldi 2006).

6 Conclusion and Future Work

This paper has presented a suite of open-source tools which we believe will be of value to the MT research community.

We have also described a new SMT decoder which can incorporate some linguistic features in a consistent and flexible framework. This new direction in research opens up many possibilities and issues that require further research and experimentation. Initial results show the potential benefit of factors for statistical machine translation, (Koehn et al. 2006) and (Koehn and Hoang 2007).

References

Axelrod, Amittai. *"Factored Language Model for Statistical Machine Translation."* MRes Thesis. Edinburgh University, 2006.

Bertoldi, Nicola, and Marcello Federico. *"A New Decoder for Spoken Language Translation Based on Confusion Networks."* Automatic Speech Recognition and Understanding Workshop (ASRU), 2005.

Bilmes, Jeff A, and Katrin Kirchhoff. *"Factored Language Models and Generalized Parallel Backoff."* HLT/NACCL, 2003.

Koehn, Philipp. *"Pharaoh: A Beam Search Decoder for Phrase-Based Statistical Machine Translation Models."* AMTA, 2004.

Koehn, Philipp, Marcello Federico, Wade Shen, Nicola Bertoldi, Ondrej Bojar, Chris Callison-Burch, Brooke Cowan, Chris Dyer, Hieu Hoang, Richard Zens, Alexandra Constantin, Christine Corbett Moran, and Evan Herbst. "*Open Source Toolkit for Statistical Machine Translation"*. Report of the 2006 Summer Workshop at Johns Hopkins University, 2006.

Koehn, Philipp, and Hieu Hoang. *"Factored Translation Models."* EMNLP, 2007.

Koehn, Philipp, Franz Josef Och, and Daniel Marcu. *"Statistical Phrase-Based Translation."* HLT/NAACL, 2003.

Och, Franz Josef. *"Minimum Error Rate Training for Statistical Machine Translation."* ACL, 2003.

Och, Franz Josef, and Hermann Ney. *"A Systematic Comparison of Various Statistical Alignment Models."* Computational Linguistics 29.1 (2003): 19-51.

Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jing Zhu. *"BLEU: A Method for Automatic Evaluation of Machine Translation."* ACL, 2002.

Shen, Wade, Richard Zens, Nicola Bertoldi, and Marcello Federico. *"The JHU Workshop 2006 Iwslt System."* International Workshop on Spoken Language Translation, 2006.

Stolcke, Andreas. *"SRILM an Extensible Language Modeling Toolkit."* Intl. Conf. on Spoken Language Processing, 2002.

Zens, Richard, and Hermann Ney. *"Efficient Phrase-Table Representation for Machine Translation with Applications to Online MT and Speech Recognition."* HLT/NAACL, 2007.

Boosting Statistical Machine Translation by Lemmatization and Linear Interpolation

Ruiqiang Zhang[1,2] and **Eiichiro Sumita**[1,2]
[1]National Institute of Information and Communications Technology
[2]ATR Spoken Language Communication Research Laboratories
2-2-2 Hikaridai, Seiika-cho, Soraku-gun, Kyoto, 619-0288, Japan
{ruiqiang.zhang,eiichiro.sumita}@atr.jp

Abstract

Data sparseness is one of the factors that degrade statistical machine translation (SMT). Existing work has shown that using morpho-syntactic information is an effective solution to data sparseness. However, fewer efforts have been made for Chinese-to-English SMT with using English morpho-syntactic analysis. We found that while English is a language with less inflection, using English lemmas in training can significantly improve the quality of word alignment that leads to yield better translation performance. We carried out comprehensive experiments on multiple training data of varied sizes to prove this. We also proposed a new effective linear interpolation method to integrate multiple homologous features of translation models.

1 Introduction

Raw parallel data need to be preprocessed in the modern phrase-based SMT before they are aligned by alignment algorithms, one of which is the well-known tool, GIZA++ (Och and Ney, 2003), for training IBM models (1-4). Morphological analysis (MA) is used in data preprocessing, by which the surface words of the raw data are converted into a new format. This new format can be lemmas, stems, parts-of-speech and morphemes or mixes of these. One benefit of using MA is to ease data sparseness that can reduce the translation quality significantly, especially for tasks with small amounts of training data.

Some published work has shown that applying morphological analysis improved the quality of SMT (Lee, 2004; Goldwater and McClosky, 2005). We found that all this earlier work involved experiments conducted on translations from highly inflected languages, such as Czech, Arabic, and Spanish, to English. These researchers also provided detailed descriptions of the effects of foreign language morpho-syntactic analysis but presented no specific results to show the effect of English morphological analysis. To the best of our knowledge, there have been no papers related to English morphological analysis for Chinese-to-English (CE) translations even though the CE translation has been the main track for many evaluation campaigns including NIST MT, IWSLT and TC-STAR, where only simple tokenization or lower-case capitalization has been applied to English preprocessing. One possible reason why English morphological analysis has been neglected may be that English is less inflected to the extent that MA may not be effective. However, we found this assumption should not be taken-for-granted.

We studied what effect English lemmatization had on CE translation. Lemmatization is shallow morphological analysis, which uses a lexical entry to replace inflected words. For example, the three words, *doing*, *did* and *done*, are replaced by one word, *do*. They are all mapped to the same Chinese translations. As a result, it eases the problem with sparse data, and retains word meanings unchanged. It is not impossible to improve word alignment by using English lemmatization.

We determined what effect lemmatization had in experiments using data from the BTEC (Paul, 2006) CSTAR track. We collected a relatively large corpus of more than 678,000 sentences. We conducted comprehensive evaluations and used multiple trans-

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 181–184,
Prague, June 2007.

lation metrics to evaluate the results. We found that our approach of using lemmatization improved both the word alignment and the quality of SMT with a small amounts of training data, and, while much work indicates that MA is useless in training large amounts of data (Lee, 2004), our intensive experiments proved that the chance to get a better MT quality using lemmatization is higher than that without it for large amounts of training data.

On the basis of successful use of lemmatization translation, we propose a new linear interpolation method by which we integrate the homologous features of translation models of the lemmatization and non-lemmatization system. We found the integrated model improved all the components' performance in the translation.

2 Moses training for system with lemmatization and without

We used Moses to carry out the expriments. Moses is the state of the art decoder for SMT. It is an extension of Pharaoh (Koehn et al., 2003), and supports factor training and decoding. Our idea can be easily implemented by Moses. We feed Moses English words with two factors: surface word and lemma. The only difference in training with lemmatization from that without is the alignment factor. The former uses Chinese surface words and English lemmas as the alignment factor, but the latter uses Chinese surface words and English surface words. Therefore, the lemmatized English is only used in word alignment. All the other options of Moses are same for both the lemmatization translation and non-lemmatization translation.

We use the tool created by (Minnen et al., 2001) to complete the morphological analysis of English. We had to make an English part-of-speech (POS) tagger that is compatible with the CLAWS-5 tagset to use this tool. We use our in-house tagset and English tagged corpus to train a statistical POS tagger by using the maximum entropy principle. Our tagset contains over 200 POS tags, most of which are consistent to the CLAWS-5. The tagger achieved 93.7% accuracy for our test set.

We use the default features defined by Pharaoh in the phrase-based log-linear models i.e., a target language model, five translation models, and one distance-based distortion model. The weighting parameters of these features were optimized in terms of BLEU by the approach of minimum error rate training (Och, 2003).

The data for training and test are from the IWSLT06 CSTAR track that uses the Basic Travel Expression Corpus (BTEC). The BTEC corpus are relatively larger corpus for travel domain. We use 678,748 Chinese/English parallel sentences as the training data in the experiments. The number of words are about 3.9M and 4.4M for Chinese and English respectively. The number of unique words for English is 28,709 before lemmatization and 24,635 after lemmatization. A 15%-20% reduction in vocabulary is obtained by the lemmatization. The test data are the one used in IWSLT06 evaluation. It contains 500 Chinese sentences. The test data of IWSLT05 are the development data for tuning the weighting parameters. Multiple references are used for computing the automatic metrics.

3 Experiments

3.1 Regular test

The purpose of the regular tests is to find what effect lemmatization has as the amount of training data increases. We used the data from the IWSLT06 CSTAR track. We started with 50,000 (50 K) of data, and gradually added more training data from a 678 K corpus to this. We applied the methods in Section 2 to train the non-lemmatized translation and lemmatized translation systems. The results are listed in Table 1. We use the alignment error rate (AER) to measure the alignment performance, and the two popular automatic metric, BLEU[1] and METEOR[2] to evaluate the translations. To measure the word alignment, we manually aligned 100 parallel sentences from the BTEC as the reference file. We use the "sure" links and the "possible" links to denote the alignments. As shown in Table 1, we found our approach improved word alignment uniformly from small amounts to large amounts of training data. The maximal AER reduction is up to 27.4% for the 600K. However, we found some mixed translation results in terms of BLEU. The lemmatized

[1] http://domino.watson.ibm.com/library/CyberDig.nsf (keyword=RC22176)

[2] http://www.cs.cmu.edu/~alavie/METEOR

Table 1: Translation results as increasing amount of training data in IWSLT06 CSTAR track

System		AER	BLEU	METEOR
50K	nonlem	0.217	0.158	0.427
	lemma	0.199	0.167	0.431
100K	nonlem	0.178	0.182	0.457
	lemma	0.177	0.188	0.463
300K	nonlem	0.150	0.223	0.501
	lemma	0.132	0.217	0.505
400K	nonlem	0.136	0.231	0.509
	lemma	0.102	0.224	0.507
500K	nonlem	0.119	0.235	0.519
	lemma	0.104	0.241	0.522
600K	nonlem	0.095	0.238	0.535
	lemma	0.069	0.248	0.536

Table 2: Statistical significance test in terms of BLEU: sys1=non-lemma, sys2=lemma

Data size	Diff(sys1-sys2)
50K	-0.092 [-0.0176,-0.0012]
100K	-0.006 [-0.0155,0.0039]
300K	0.0057 [-0.0046,0.0161]
400K	0.0074 [-0.0023,0.0174]
500K	-0.0054 [-0.0139,0.0035]
600K	-0.0103 [-0.0201,-0.0006]

translations did not outperform the non-lemmatized ones uniformly. They did for small amounts of data, i.e., 50 K and 100 K, and for large amounts, 500 K and 600 K. However, they failed for 300 K and 400 K.

The translations were under the statistical significance test by using the *bootStrap* scripts[3]. The results giving the medians and confidence intervals are shown in Table 2, where the numbers indicate the median, the lower and higher boundary at 95% confidence interval. we found the *lemma* systems were confidently better than the *nonlem* systems for the 50K and 600K, but didn't for other data sizes.

This experiments proved that our proposed approach improved the qualities of word alignments that lead to the translation improvement for the 50K, 100K, 500K and 600K. In particular, our results revealed large amounts of data of 500 K and 600 K was improved by the lemmatization while it has been found impossible in most published results. However, data of 300 K and 400 K worsen translations achieved by the lemmatization[4]. In what follows, we discuss a method of random sampling of creating multiple corpora of varied sizes to see robustness of our approach and re-investigate the results of the 300K and 400K.

Table 3: Competitive scores (BLEU) for non-lemmatization and lemmatization using randomly extracted corpora

System	100K	300K	400K	600K	total
lemma	10/11	5.5/11	6.5/11	5/7	27/40
nonlem	1/11	5.5/11	4.5/11	2/7	13/40

3.2 Random sampling test

In this section, we use a method of random extraction to generate new multiple training data for each corpus of one definite size. The new data are extracted from the whole corpus of 678 K randomly. We generate ten new corpora for 100 K, 300 K, and 400 K data and six new corpora for the 678 K data. Thus, we create eleven and seven corpora of varied sizes if the corpora in the last experiments are counted. We use the same method as in Section 2 for each generated corpus to construct systems to compare non-lemmatization and lemmatization. The systems are evaluated again using the same test data. The results are listed in Table 3 and Figure 1. Table 3 shows the "scoreboard" of non-lemmatized and lemmatized results in terms of BLEU. If its score for the *lemma* system is higher than that for the *nonlem* system, the former earns one point; if equal, each earns 0.5; otherwise, the *nonlem* earns one point. As we can see from the table, the results for the *lemma* system are better than those for the *nonlem* system for the 100K in 10 of the total 11 corpora. Of the total 40 random corpora, the *lemma* systems outperform the *nonlem* systems in 27 times.

By analyzing the results from Tables 1 and 3, we can arrive at some conclusions. The *lemma* systems outperform the *nonlem* for training corpora less than

[3] http://projectile.is.cs.cmu.edu/research/public/tools/bootStrap/tutorial.htm

[4] while the results was not confident by statistical significance test, the medians of 300K and 400K were lowered by the lemmatization

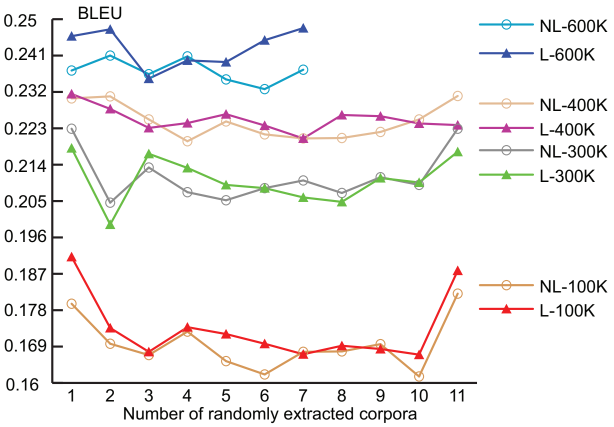


Figure 1: Bleu scores for randomly extracted corpora

100 K. The BLEU score favors the *lemma* system overwhelmingly for this size. When the amount of training data is increased up to 600 K, the *lemma* still beat the *nonlem* system in most tests while the number of success by the *nonlem* system increases. This random test, as a complement to the last experiment, reveals that the *lemma* either performs the same or better than the *nonlem* system for training data of any size. Therefore, the *lemma* system is slightly better than the *nonlem* in general.

Figure 1 illustrates the BLEU scores for the "lemma(L)" and "nonlem(NL)" systems for randomly extracted corpora. A higher number of points is obtained by the *lemma* system than the *nonlem* for each corpus.

4 Effect of linear interpolation of features

We generated translation models for lemmatization translation and non-lemmatization translation. We found some features of the translation models could be added linearly. For example, phrase translation model $p(e|f)$ can be calculated as,

$$p(e|f) = \alpha_1 p_l(e|f) + \alpha_2 p_{nl}(e|f)$$

where $p_l(e|f)$ and $p_{nl}(e|f)$ is the phrase translation models corresponding to the lemmatization system and non-lemma system. $\alpha_1 + \alpha_2 = 1$. αs can be obtained by maximizing likelihood or BLEU scores of a development data. But we used the same values for all the α. $p(e|f)$ is the phrase translation model after linear interpolation. Besides the phrase translation model, we used this approach to integrate the three other features: phrase inverse probability, lexical probability, and lexical inverse probability. We tested this integration using the open track of IWSLT 2006, a small task track. The BLEU scores are shown in Table 4. An improvement over both of the systems were observed.

Table 4: Effect of linear interpolation

	lemma	nonlemma	interpolation
open track	0.1938	0.1993	0.2054

5 Conclusions

We proposed a new approach of using lemmatization and linear interpolation of homologous features in SMT. The principal idea is to use lemmatized English for the word alignment. Our approach was proved effective for the BTEC Chinese to English translation. It is significant in particular that we have target language, English, as the lemmatized object because it is less usual in SMT. Nevertheless, we found our approach significantly improved word alignment and qualities of translations.

References

Sharon Goldwater and David McClosky. 2005. Improving statistical MT through morphological analysis. In *Proceedings of HLT/EMNLP*, pages 676–683, Vancouver, British Columbia, Canada, October.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *HLT-NAACL 2003: Main Proceedings*, pages 127–133.

Young-Suk Lee. 2004. Morphological analysis for statistical machine translation. In *HLT-NAACL 2004: Short Papers*, pages 57–60, Boston, Massachusetts, USA.

Guido Minnen, John Carroll, and Darren Pearce. 2001. Applied morphological processing of english. *Natural Language Engineering*, 7(3):207–223.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *ACL 2003*, pages 160–167.

Michael Paul. 2006. Overview of the IWSLT 2006 Evaluation Campaign. In *Proc. of the IWSLT*, pages 1–15, Kyoto, Japan.

Extractive Summarization Based on Event Term Clustering

Maofu Liu[1,2], **Wenjie Li**[1], **Mingli Wu**[1] **and Qin Lu**[1]

[1]Department of Computing
The Hong Kong Polytechnic University
{csmfliu, cswjli, csmlwu, csluqin}@comp.polyu.edu.hk

[2]College of Computer Science and Technology
Wuhan University of Science and Technology
mfliu_china@hotmail.com

Abstract

Event-based summarization extracts and organizes summary sentences in terms of the events that the sentences describe. In this work, we focus on semantic relations among event terms. By connecting terms with relations, we build up event term graph, upon which relevant terms are grouped into clusters. We assume that each cluster represents a topic of documents. Then two summarization strategies are investigated, i.e. selecting one term as the representative of each topic so as to cover all the topics, or selecting all terms in one most significant topic so as to highlight the relevant information related to this topic. The selected terms are then responsible to pick out the most appropriate sentences describing them. The evaluation of clustering-based summarization on DUC 2001 document sets shows encouraging improvement over the well-known PageRank-based summarization.

1 Introduction

Event-based extractive summarization has emerged recently (Filatova and Hatzivassiloglou, 2004). It extracts and organizes summary sentences in terms of the events that sentences describe.

We follow the common agreement that event can be formulated as “[Who] did [What] to [Whom] [When] and [Where]” and “did [What]” denotes the key element of an event, i.e. the action within the formulation. We approximately define the verbs and action nouns as the event terms which can characterize or partially characterize the event occurrences.

Most existing event-based summarization approaches rely on the statistical features derived from documents and generally associated with single events, but they neglect the relations among events. However, events are commonly related with one another especially when the documents to be summarized are about the same or very similar topics. Li et al (2006) report that the improved performance can be achieved by taking into account of event distributional similarities, but it does not benefit much from semantic similarities. This motivated us to further investigate whether event-based summarization can take advantage of the semantic relations of event terms, and most importantly, how to make use of those relations. Our idea is grouping the terms connected by the relations into the clusters, which are assumed to represent some topics described in documents.

In the past, various clustering approaches have been investigated in document summarization. Hatzivassiloglou et al (2001) apply clustering method to organize the highly similar paragraphs into tight clusters based on primitive or composite features. Then one paragraph per cluster is selected to form the summary by extraction or by reformulation. Zha (2002) uses spectral graph clustering algorithm to partition sentences into topical groups. Within each cluster, the saliency scores of terms and sentences are calculated using mutual reinforcement principal, which assigns high salience scores to the sentences that contain many terms with high salience scores. The sentences and key phrases are selected by their saliency scores to generate the summary. The similar work based on topic or event is also reported in (Guo and Stylios, 2005).

The granularity of clustering units mentioned above is rather coarse, either sentence or paragraph. In this paper, we define event term as clustering

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 185–188,
Prague, June 2007.

unit and implement a clustering algorithm based on semantic relations. We extract event terms from documents and construct the event term graph by linking terms with the relations. We then regard a group of closely related terms as a topic and make the following two alterative assumptions:

(1) If we could find the most significant topic as the main topic of documents and select all terms in it, we could summarize the documents with this main topic.

(2) If we could find all topics and pick out one term as the representative of each topic, we could obtain the condensed version of topics described in the documents.

Based on these two assumptions, a set of cluster ranking, term selection and ranking and sentence extraction strategies are developed. The remainder of this paper is organized as follows. Section 2 introduces the proposed extractive summarization approach based on event term clustering. Section 3 presents experiments and evaluations. Finally, Section 4 concludes the paper.

2 Summarization Based on Event Term Clustering

2.1 Event Term Graph

We introduce VerbOcean (Chklovski and Pantel, 2004), a broad-coverage repository of semantic verb relations, into event-based summarization. Different from other thesaurus like WordNet, VerbOcean provides five types of semantic verb relations at finer level. This just fits in with our idea to introduce event term relations into summarization. Currently, only the stronger-than relation is explored. When two verbs are similar, one may denote a more intense, thorough, comprehensive or absolute action. In the case of change-of-state verbs, one may denote a more complete change. This is identified as the stronger-than relation in (Timothy and Patrick, 2004). In this paper, only stronger-than is taken into account but we consider extending our future work with other applicable relations types.

The event term graph connected by term semantic relations is defined formally as $G=(V,E)$, where V is a set of event terms and E is a set of relation links connecting the event terms in V. The graph is directed if the semantic relation has the characteristic of the asymmetric. Otherwise, it is undirected. Figure 1 shows a sample of event term graph built from one DUC 2001 document set. It is a directed graph as the *stronger-than* relation in VerbOcean exhibits the conspicuous asymmetric characteristic. For example, “fight” means to attempt to harm by blows or with weapons, while “resist” means to keep from giving in. Therefore, a directed link from “fight” to “resist” is shown in the following Figure 1.

Relations link terms together and form the event term graph. Based upon it, term significance is evaluated and in turn sentence is judged whether to be extracted in the summary.

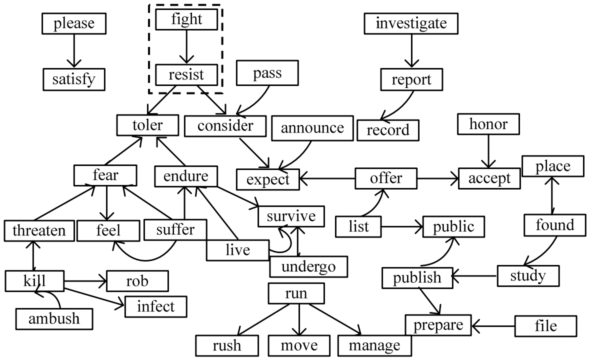


Figure 1. Terms connected by semantic relations

2.2 Event Term Clustering

Note that in Figure 1, some linked event terms, such as “kill”, “rob”, “threaten” and “infect”, are semantically closely related. They may describe the same or similar topic somehow. In contrast, “toler”, “resist” and “fight” are clearly involved in another topic; although they are also reachable from “kill”. Based on this observation, a clustering algorithm is required to group the similar and related event terms into the cluster of the topic.

In this work, event terms are clustered by the DBSCAN, a density-based clustering algorithm proposed in (Easter et al, 1996). The key idea behind it is that for each term of a cluster the neighborhood of a given radius has to contain at least a minimum number of terms, i.e. the density in the neighborhood has to exceed some threshold. By using this algorithm, we need to figure out appropriate values for two basic parameters, namely, *Eps* (denoting the searching radius from each term) and *MinPts* (denoting the minimum number of terms in the neighborhood of the term). We assign one semantic relation step to *Eps* since there is no clear distance concept in the event term graph. The value of *Eps* is experimentally set in our experiments. We also make some modification on Easter's DBSCAN in order to accommodate to our task.

Figure 2 shows the seven term clusters generated by the modified DBSCAN clustering algorithm from the graph in Figure 1. We represent each cluster by the starting event term in bold font.

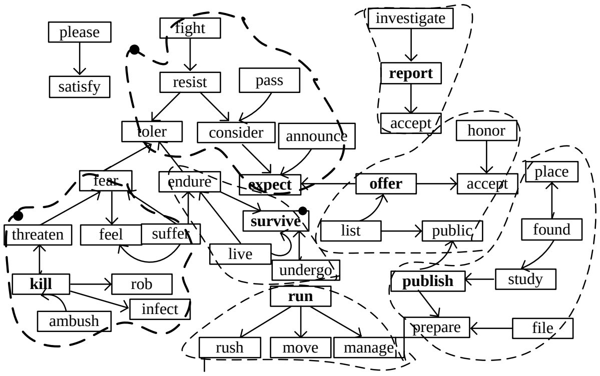


Figure 2. Term clusters generated from Figure 1

2.3 Cluster Ranking

The significance of the cluster is calculated by

$$sc(C_i) = \sum_{t \in C_i} d_t \Big/ \sum_{C_i \in C} \sum_{t \in C_i} d_t$$

where d_t is the degree of the term t in the term graph. C is the set of term clusters obtained by the modified DBSCAN clustering algorithm and C_i is the *ith* one. Obviously, the significance of the cluster is calculated from global point of view, i.e. the sum of the degree of all terms in the same cluster is divided by the total degree of the terms in all clusters.

2.4 Term Selection and Ranking

Representative terms are selected according to the significance of the event terms calculated within each cluster (i.e. from local point of view) or in all clusters (i.e. from global point of view) by

LOCAL: $st(t) = d_t \Big/ \sum_{t \in c_i} d_t$ or

GLOBAL: $st(t) = d_t \Big/ \sum_{c_i \in C} \sum_{t \in c_i} d_t$

Then two strategies are developed to select the representative terms from the clusters.

(1) One Cluster All Terms (**OCAT**) selects all terms within the first rank cluster. The selected terms are then ranked according to their significance.

(2) One Term All Cluster (**OTAC**) selects one most significant term from each cluster. Notice that because terms compete with each other within clusters, it is not surprising to see $st(t_1) < st(t_2)$ even when $sc(c_1) > sc(c_2)$, $(t_1 \in c_1, t_2 \in c_2)$. To address this problem, the representative terms are ranked according to the significance of the clusters they belong to.

2.5 Sentence Evaluation and Extraction

A representative event term may associate to more than one sentence. We extract only one of them as the description of the event. To this end, sentences are compared according to the significance of the terms in them. **MAX** compares the maximum significance scores, while **SUM** compares the sum of the significance scores. The sentence with either higher MAX or SUM wins the competition and is picked up as a candidate summary sentence. If the sentence in the first place has been selected by another term, the one in the second place is chosen. The ranks of these candidates are the same as the ranks of the terms they are selected for. Finally, candidate sentences are selected in the summary until the length limitation is reached.

3 Experiments

We evaluate the proposed approaches on DUC 2001 corpus which contains 30 English document sets. There are 431 event terms on average in each document set. The automatic evaluation tool, ROUGE (Lin and Hovy, 2003), is run to evaluate the quality of the generated summaries (200 words in length). The tool presents three values including unigram-based ROUGE-1, bigram-based ROUGE-2 and ROUGE-W which is based on longest common subsequence weighted by the length.

Google's PageRank (Page and Brin, 1998) is one of the most popular ranking algorithms. It is also graph-based and has been successfully applied in summarization. Table 1 lists the result of our implementation of PageRank based on event terms. We then compare it with the results of the event term clustering-based approaches illustrated in Table 2.

	PageRank
ROUGE-1	0.32749

ROUGE-2	0.05670
ROUGE-W	0.11500

Table 1. Evaluations of PageRank-based Summarization

LOCAL+OTAC	MAX	SUM
ROUGE-1	0.32771	0.33243
ROUGE-2	0.05334	0.05569
ROUGE-W	0.11633	0.11718
GLOBAL+OTAC	MAX	SUM
ROUGE-1	0.32549	0.32966
ROUGE-2	0.05254	0.05257
ROUGE-W	0.11670	0.11641
LOCAL+OCAT	MAX	SUM
ROUGE-1	0.33519	0.33397
ROUGE-2	0.05662	0.05869
ROUGE-W	0.11917	0.11849
GLOBAL+OCAT	MAX	SUM
ROUGE-1	0.33568	0.33872
ROUGE-2	0.05506	0.05933
ROUGE-W	0.11795	0.12011

Table 2. Evaluations of Clustering-based Summarization

The experiments show that both assumptions are reasonable. It is encouraging to find that our event term clustering-based approaches could outperform the PageRank-based approach. The results based on the second assumption are even better. This suggests indeed there is a main topic in a DUC 2001 document set.

4 Conclusion

In this paper, we put forward to apply clustering algorithm on the event term graph connected by semantic relations derived from external linguistic resource. The experiment results based on our two assumptions are encouraging. Event term clustering-based approaches perform better than PageRank-based approach. Current approaches simply utilize the degrees of event terms in the graph. In the future, we would like to further explore and integrate more information derived from documents in order to achieve more significant results using the event term clustering-based approaches.

Acknowledgments

The work described in this paper was fully supported by a grant from the Research Grants Council of the Hong Kong Special Administrative Region, China (Project No. PolyU5181/03E).

References

Chin-Yew Lin and Eduard Hovy. 2003. Automatic Evaluation of Summaries using *N*-gram Cooccurrence Statistics. In Proceedings of HLT/ NAACL 2003, pp71-78.

Elena Filatova and Vasileios Hatzivassiloglou. 2004. Event-based Extractive Summarization. In Proceedings of ACL 2004 Workshop on Summarization, pp104-111.

Hongyuan Zha. 2002. Generic Summarization and keyphrase Extraction using Mutual Reinforcement Principle and Sentence Clustering. In Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval, 2002. pp113-120.

Lawrence Page and Sergey Brin, Motwani Rajeev and Winograd Terry. 1998. The PageRank CitationRanking: Bring Order to the Web. Technical Report,Stanford University.

Martin Easter, Hans-Peter Kriegel, Jörg Sander, et al. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Menlo Park, CA, 1996. 226-231.

Lawrence Page, Sergey Brin, Rajeev Motwani and Terry Winograd. 1998. The PageRank CitationRanking: Bring Order to the Web. Technical Report,Stanford University.

Timothy Chklovski and Patrick Pantel. 2004. VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In Proceedings of Conference on Empirical Methods in Natural Language Processing, 2004.

Vasileios Hatzivassiloglou, Judith L. Klavans, Melissa L. Holcombe, et al. 2001. Simfinder: A Flexible Clustering Tool for Summarization. In Workshop on Automatic Summarization, NAACL, 2001.

Wenjie Li, Wei Xu, Mingli Wu, et al. 2006. Extractive Summarization using Inter- and Intra-Event Relevance. In Proceedings of ACL 2006, pp369-376.

Yi Guo and George Stylios. 2005. An intelligent summarization system based on cognitive psychology. Journal of Information Sciences, Volume 174, Issue 1-2, Jun. 2005, pp1-36.

Machine Translation between Turkic Languages

A. Cüneyd TANTUĞ
Istanbul Technical University
Istanbul, Turkey
tantug@itu.edu.tr

Eşref ADALI
Istanbul Technical University
Istanbul, Turkey
adali@itu.edu.tr

Kemal OFLAZER
Sabanci University
Istanbul, Turkey
oflazer@sabanciuniv.edu

Abstract

We present an approach to MT between Turkic languages and present results from an implementation of a MT system from Turkmen to Turkish. Our approach relies on ambiguous lexical and morphological transfer augmented with target side rule-based repairs and rescoring with statistical language models.

1 Introduction

Machine translation is certainly one of the toughest problems in natural language processing. It is generally accepted however that machine translation between close or related languages is simpler than full-fledged translation between languages that differ substantially in morphological and syntactic structure. In this paper, we present a machine translation system from Turkmen to Turkish, both of which belong to the Turkic language family. Turkic languages essentially exhibit the same characteristics at the morphological and syntactic levels. However, except for a few pairs, the languages are not mutually intelligible owing to substantial divergences in their lexicons possibly due to different regional and historical influences. Such divergences at the lexical level along with many but minor divergences at morphological and syntactic levels make the translation problem rather non-trivial. Our approach is based on essentially morphological processing, and direct lexical and morphological transfer, augmented with substantial multi-word processing on the source language side and statistical processing on the target side where data for statistical language modelling is more readily available.

2 Related Work

Studies on machine translation between close languages are generally concentrated around certain Slavic languages (e.g., Czech→Slovak, Czech→Polish, Czech→Lithuanian (Hajic et al., 2003)) and languages spoken in the Iberian Peninsula (e.g., Spanish↔Catalan (Canals et al., 2000), Spanish↔Galician (Corbi-Bellot et al., 2003) and Spanish↔Portugese (Garrido-Alenda et al., 2003). Most of these implementations use similar modules: a morphological analyzer, a part-of-speech tagger, a bilingual transfer dictionary and a morphological generator. Except for the Czech→Lithuanian system which uses a shallow parser, syntactic parsing is not necessary in most cases because of the similarities in word orders. Also, the lexical semantic ambiguity is usually preserved so, none of these systems has any module for handling the lexical ambiguity. For Turkic languages, Hamzaoğlu (1993) has developed a system from Turkish to Azerbaijani, and Altıntaş (2000) has developed a system from Turkish to Crimean Tatar.

3 Turkic Languages

Turkic languages, spoken by more than 180 million people, constitutes subfamily of Ural-Altaic languages and includes languages like Turkish, Azerbaijani, Turkmen, Uzbek, Kyrghyz, Kazakh, Tatar, Uyghur and many more. All Turkic languages have very productive inflectional and derivational agglutinative morphology. For example the Turkish word *evlerimizden* has three inflectional morphemes attached to a noun root *ev* (house), for the plural form with second person plural possessive agreement and ablative case:

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 189–192,
Prague, June 2007. ©2007 Association for Computational Linguistics

```
evlerimizden    (from our houses)
ev+ler+imiz+den
ev+Noun+A3pl+P1sg+Abl
```

All Turkic languages exhibit SOV constituent order but depending on discourse requirements, constituents can be in any order without any substantial formal constraints. Syntactic structures between Turkic languages are more or less parallel though there are interesting divergences due to mismatches in multi-word or idiomatic constructions.

4 Approach

Our approach is based on a direct morphological transfer with some local multi-word processing on the source language side, and statistical disambiguation on the target language side. The main steps of our model are:

1. Source Language (SL) Morphological Analysis
2. SL Morphological Disambiguation
3. Multi-Word Unit (MWU) Recognizer
4. Morphological Transfer
5. Root Word Transfer
6. Statistical Disambiguation and Rescoring (SLM)
7. Sentence Level Rules (SLR)
8. Target Language (TL) Morphological Generator

Steps other than 3, 6 and 7 are the minimum requirements for a direct morphological translation model (henceforth, the baseline system). The MWU Recognizer, SLM and SLR modules are additional modules for the baseline system to improve the translation quality.

Source language morphological analysis may produce multiple interpretation of a source word, and usually, depending on the ambiguities brought about by multiple possible segmentations into root and suffixes, there may be different root words of possibly different parts-of-speech for the same word form. Furthermore, each root word thus produced may map to multiple target root words due to word sense ambiguity. Hence, among all possible sentences that can be generated with these ambiguities, the most probable one is selected by using various types of SLMs that are trained on target language corpora annotated with disambiguated roots and morphological features.

MWU processing in Turkic languages involves more than the usual lexicalized collocations and involves detection of mostly unlexicalized intra-word morphological patterns (Oflazer et al., 2004). Source MWUs are recognized and marked during source analysis and the root word transfer module maps these either to target MWU patterns, or directly translates when there is a divergence.

Morphological transfer is implemented by a set of rules hand-crafted using the contrastive knowledge between the selected language pair.

Although the syntactic structures are very similar between Turkic languages, there are quite many minor situations where target morphological features marking features such as subject-verb agreement have to be recovered when such features are not present in the source. Furthermore, occasionally certain phrases have to be rearranged. Finally, a morphological generator produces the surface forms of the lexical forms in the sentence.

5 Turkmen to Turkish MT System

The first implementation of our approach is from Turkmen to Turkish. A general diagram of our MT system is presented in Figure 1. The morphological analysis on the Turkmen side is performed by a two-level morphological analyzer developed using Xerox finite state tools (Tantuğ et al., 2006). It takes a Turkmen word and produces all possible morphological interpretations of that word. A simple experiment on our test set indicates that the average Turkmen word gets about 1.55 analyses. The multi-word recognition module operates on the output of the morphological analyzer and wherever applicable, combines analyses of multiple tokens into a new analysis with appropriate morphological features. One side effect of multi-word processing is a small reduction in morphological ambiguity, as when such units are combined, the remaining morphological interpretations for these tokens are deleted.

The actual transfer is carried out by transferring the morphological structures and word roots from the source language to the target language maintaining any ambiguity in the process. These are implemented with finite state transducers that are compiled from replace rules written in the Xerox regular expression language.[1] A very simple example of this transfer is shown in Figure 2.[2]

[1] The current implementation employs 28 replace rules for morphological feature transfer and 19 rules for sentence level processing.

[2] +Pos:Positive polarity, +A3sg: 3^{rd} person singular agreement, +Inf1,+Inf2: infinitive markers, +P3sg, +Pnon: possessive agreement markers, +Nom,+Acc: Nominative and ac-

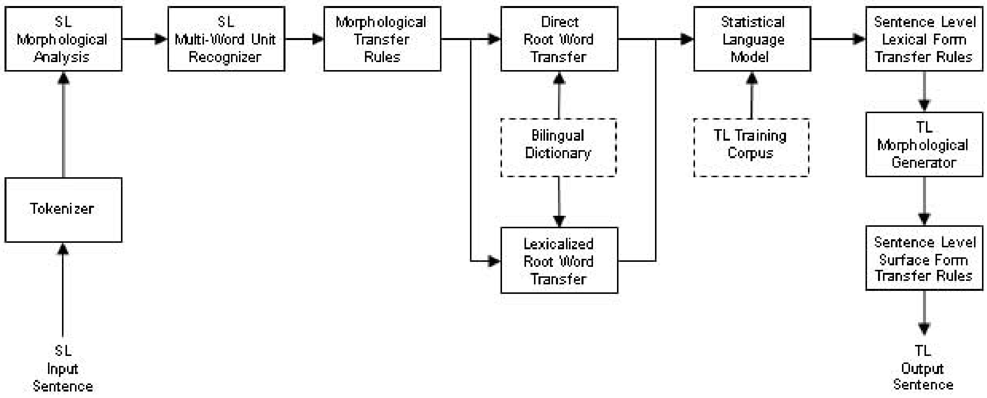

SL Morphological Analysis
SL Multi-Word Unit Recognizer
Morphological Transfer Rules
Direct Root Word Transfer
Statistical Language Model
Sentence Level Lexical Form Transfer Rules
Tokenizer
Bilingual Dictionary
TL Training Corpus
TL Morphological Generator
Lexicalized Root Word Transfer
Sentence Level Surface Form Transfer Rules
SL Input Sentence
TL Output Sentence

Figure 1: Main blocks of the translation system

```
ösmegi
↓
```
Source Morphological Analysis
```
↓
ös+Verb+Pos^DB+Noun+Inf1+A3sg+P3sg+Nom
ös+Verb+Pos^DB+Noun+Inf1+A3sg+Pnon+Acc
↓
```
Source-to-Target Morphological Feature Transfer
```
↓
ös+Verb+Pos^DB+Noun+Inf2+A3sg+P3sg+Nom
ös+Verb+Pos^DB+Noun+Inf2+A3sg+Pnon+Acc
↓
```
Source-to-Target Root word Transfer
```
↓
ilerle+Verb+Pos^DB+Noun+Inf2+A3sg+P3sg+Nom
ilerle+Verb+Pos^DB+Noun+Inf2+A3sg+Pnon+Acc
büyü+Verb+Pos^DB+Noun+Inf2+A3sg+P3sg+Nom
büyü+Verb+Pos^DB+Noun+Inf2+A3sg+Pnon+Acc
↓
```
Target Morphological Generation

↓

`ilerlemesi` (the progress of (something))
`ilerlemeyi` (the progress (as direct object))
`büyümesi` (the growth of (something))
`büyümeyi` (the growth (as direct object))

Figure 2: Word transfer

In this example, once the morphological analysis is produced, first we do a morphological feature transfer mapping. In this case, the only interesting mapping is the change of the infinitive marker. The source root verb is then ambiguously mapped to two verbs on the Turkish side. Finally, the Turkish surface form is generated by the morphological generator. Note that all the morphological processing details such as vowel harmony resolution (a morphographemic process common to all Turkic languages though not in identical ways) are localized to morphological generation.

Root word transfer is also based on a large transducer compiled from bilingual dictionaries which contain many-to-many mappings. During mapping this transducer takes into account the source root word POS.[3] In some rare cases, mapping the word root is not sufficient to generate a legal Turkish lexical structure, as sometimes a required feature on the target side may not be explicitly available on the source word to generate a proper word. In order to produce the correct mapping in such cases, some additional lexicalized rules look at a wider context and infer any needed features.

While the output of morphological feature transfer module is usually unambiguous, ambiguity arises during the root word transfer phase. We attempt to resolve this ambiguity on the target language side using statistical language models. This however presents additional difficulties as any statistical language model for Turkish (and possibly other Turkic languages) which is built by using the surface forms suffers from data sparsity problems. This is due to agglutinative morphology whereby a root word may give rise to too many inflected forms (about a hundred inflected forms for nouns and much more for verbs; when productive derivations are considered these numbers grow substantially!). Therefore, instead of building statistical language models on full word forms, we work with morphologically analyzed and disambiguated target language corpora. For example, we use a language model that is only based on the (disambiguated) root words to disambiguate ambiguous root words that arise from root

cusative case markers.

[3] Statistics on the test set indicate that on the average each source language root word maps to about 2 target language root words.

word transfer. We also employ a language model which is trained on the last set of inflectional features of morphological parses (hence does not involve any root words.)

Although word-by-word translation can produce reasonably high quality translations, but in many cases, it is also the source of many translation errors. To alleviate the shortcomings of the word-by-word translation approach, we resort to a series of rules that operate across the whole sentence. Such rules operate on the lexical and surface representation of the output sentence. For example, when the source language is missing a subject agreement marker on a verb, this feature can not be transferred to the target language and the target language generator will fail to generate the appropriate word. We use some simple heuristics that try to recover the agreement information from any overt pronominal subject in nominative case, and that failing, set the agreement to 3^{rd} person singular. Some sentence level rules require surface forms because this set of rules usually make orthographic changes affected by previous word forms. In the following example, suitable variants of the clitics *de* and *mi* must be selected so that vowel harmony with the previous token is preserved.

o d**e** gördü m**i**? → o d**a** gördü m**ü**?
(did he see too?)

A wide-coverage Turkish morphological analyzer (Oflazer, 1994) made available to be used in reverse direction to generate the surface forms of the translations.

6 Results and Evaluation

We have tracked the progress of our changes to our system using the BLEU metric (Papineni et al., 2004), though it has serious drawbacks for agglutinative and free constituent order languages.

The performance of the baseline system (all steps above, except 3, 6, and 7) and systems with additional modules are given in Table 1 for a set of 254 Turkmen sentences with 2 reference translations each. As seen in the table, each module contributes to the performance of the baseline system. Furthermore, a manual investigation of the outputs indicates that the actual quality of the translations is higher than the one indicated by the BLEU score.[4] The errors mostly stem from the statical language models not doing a good job at selecting the right root words and/or the right morphological features.

System	BLEU Score
Baseline	26.57
Baseline + MWU	28.45
Baseline + MWU + SLM	31.37
Baseline + MWU + SLM + SLR	33.34

Table 1: BLEU Scores

7 Conclusions

We have presented an MT system architecture between Turkic languages using morphological transfer coupled with target side language modelling and results from a Turkmen to Turkish system. The results are quite positive but there is quite some room for improvement. Our current work involves improving the quality of our current system as well as expanding this approach to Azerbaijani and Uyghur.

Acknowledgments

This work was partially supported by Project 106E048 funded by The Scientific and Technical Research Council of Turkey. Kemal Oflazer acknowledges the kind support of LTI at Carnegie Mellon University, where he was a sabbatical visitor during the academic year 2006 – 2007.

References

A. Cüneyd Tantuğ, Eşref Adalı, Kemal Oflazer. 2006. Computer Analysis of the Turkmen Language Morphology. *FinTAL, Lecture Notes in Computer Science*, 4139:186-193.

A. Garrido-Alenda et al. 2003. Shallow Parsing for Portuguese-Spanish Machine Translation. *in TASHA 2003: Workshop on Tagging and Shallow Processing of Portuguese*, Lisbon, Portugal.

A. M. Corbi-Bellot et al. 2005. An open-source shallow-transfer machine translation engine for the Romance languages of Spain. *in 10th EAMT conference "Practical applications of machine translation"*, Budapest, Hungary.

Jan Hajic, Petr Homola, Vladislav Kubon. 2003. A simple multilingual machine translation system. *MT Summit IX.*

İlker Hamzaoğlu. 1993. Machine translation from Turkish to other Turkic languages and an implementation for the Azeri language. *MSc Thesis, Bogazici University, Istanbul*.

Kemal Altıntaş. 2000. Turkish to Crimean Tatar Machine Translation System. *MSc Thesis, Bilkent University, Ankara*.

Kemal Oflazer. 1994. Two-level description of Turkish morphology. *Literary and Linguistic Computing*, 9(2).

Kemal Oflazer, Özlem Çetinoğlu, Bilge Say. 2004. Integrating Morphology with Multi-word Expression Processing in Turkish. *The ACL 2004 Workshop on Multiword Expressions: Integrating Processing*.

Kishore Papineni et al. 2002. BLEU : A Method for Automatic Evaluation of Machine Translation. *Association of Computational Linguistics, ACL'02*.

Raul Canals-Marote et al. 2000. interNOSTRUM: a Spanish-Catalan Machine Translation System. *Machine Translation Review*, 11:21-25.

[4] There are many translations which preserve the same meaning with the references but get low BLEU scores.

Measuring Importance and Query Relevance in Topic-focused Multi-document Summarization

Surabhi Gupta and **Ani Nenkova** and **Dan Jurafsky**
Stanford University
Stanford, CA 94305
surabhi@cs.stanford.edu, {anenkova,jurafsky}@stanford.edu

Abstract

The increasing complexity of summarization systems makes it difficult to analyze exactly which modules make a difference in performance. We carried out a principled comparison between the two most commonly used schemes for assigning importance to words in the context of *query focused multi-document summarization*: raw frequency (word probability) and log-likelihood ratio. We demonstrate that the advantages of log-likelihood ratio come from its known distributional properties which allow for the identification of *a set* of words that in its entirety defines the aboutness of the input. We also find that LLR is more suitable for query-focused summarization since, unlike raw frequency, it is more sensitive to the integration of the information need defined by the user.

1 Introduction

Recently the task of multi-document summarization in response to a complex user query has received considerable attention. In *generic summarization*, the summary is meant to give an overview of the information in the documents. By contrast, when the summary is produced in response to a user query or topic (*query-focused, topic-focused,* or generally *focused* summary), the topic/query determines what information is appropriate for inclusion in the summary, making the task potentially more challenging.

In this paper we present an analytical study of two questions regarding aspects of the topic-focused scenario. First, two estimates of importance on words have been used very successfully both in generic and query-focused summarization: *frequency* (Luhn, 1958; Nenkova et al., 2006; Vanderwende et al., 2006) and *loglikelihood ratio* (Lin and Hovy, 2000; Conroy et al., 2006; Lacatusu et al., 2006). While both schemes have proved to be suitable for summarization, with generally better results from log-likelihood ratio, no study has investigated in what respects and by how much they differ. Second, there are many little-understood aspects of the differences between generic and query-focused summarization. For example, we'd like to know if a particular word weighting scheme is more suitable for focused summarization than others. More significantly, previous studies show that generic and focused systems perform very similarly to each other in query-focused summarization (Nenkova, 2005) and it is of interest to find out why.

To address these questions we examine the two weighting schemes: raw frequency (or word probability estimated from the input), and log-likelihood ratio (LLR) and two of its variants. These metrics are used to assign importance to individual content words in the input, as we discuss below.

Word probability $R(w) = \frac{n}{N}$, where n is the number of times the word w appeared in the input and N is the total number of words in the input.

Log-likelihood ratio (LLR) The likelihood ratio λ (Manning and Schutze, 1999) uses a background corpus to estimate the importance of a word and it is proportional to the mutual information between a word w and the input to be summarized; $\lambda(w)$ is defined as the ratio between the probability (under a binomial distribution) of observing w in the input and the background corpus assuming equal probability of occurrence of w in both and the probability of the data assuming different probabilities for w in the input and the background corpus.

LLR with cut-off (LLR(C)) A useful property of the log-likelihood ratio is that the quantity

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 193–196,
Prague, June 2007. ©2007 Association for Computational Linguistics

$-2\log(\lambda)$ is asymptotically well approximated by χ^2 distribution. A word appears in the input significantly more often than in the background corpus when $-2\log(\lambda) > 10$. Such words are called signature terms in Lin and Hovy (2000) who were the first to introduce the log-likelihood weighting scheme for summarization. Each descriptive word is assigned an equal weight and the rest of the words have a weight of zero:

$R(w) = 1$ if $(-2\log(\lambda(w)) > 10)$, 0 otherwise.

This weighting scheme has been adopted in several recent generic and topic-focused summarizers (Conroy et al., 2006; Lacatusu et al., 2006).

LLR(CQ) The above three weighting schemes assign a weight to words regardless of the user query and are most appropriate for generic summarization. When a user query *is* available, it should inform the summarizer to make the summary more focused. In Conroy et al. (2006) such query sensititivity is achieved by augmenting LLR(C) with all content words from the user query, each assigned a weight of 1 equal to the weight of words defined by LLR(C) as topic words from the input to the summarizer.

2 Data

We used the data from the 2005 Document Understanding Conference (DUC) for our experiments. The task is to produce a 250-word summary in response to a topic defined by a user for a total of 50 topics with approximately 25 documents for each marked as relevant by the topic creator. In computing LLR, the remaining 49 topics were used as a background corpus as is often done by DUC participants. A sample topic (d301) shows the complexity of the queries:

> Identify and describe types of organized crime that crosses borders or involves more than one country. Name the countries involved. Also identify the perpetrators involved with each type of crime, including both individuals and organizations if possible.

3 The Experiment

In the summarizers we compare here, the various weighting methods we describe above are used to assign importance to individual content words in the input. The weight or importance of a sentence S in the input is defined as

$$Weight_{R(S)} = \sum_{w \in S} R(w) \qquad (1)$$

where $R(w)$ assigns a weight for each word w.

	GENERIC	FOCUSED
Frequency	0.11972 (0.11168–0.12735)	**0.11795** (0.11010–0.12521)
LLR	*0.11223* (0.10627–0.11873)	0.11600 (0.10915–0.12281)
LLR(C)	*0.11949* (0.11249–0.12724)	0.12201 (0.11507–0.12950)
LLR(CQ)	*not app*	**0.12546** (.11884–.13247)

Table 1: SU4 ROUGE recall (and 95% confidence intervals) for runs on the entire input (GENERIC) and on relevant sentences (FOCUSED).

For GENERIC summarization, the top scoring sentences in the input are taken to form a generic extractive summary. In the computation of sentence importance, only nouns, verbs, adjectives and adverbs are considered and a short list of light verbs are excluded: “has, was, have, are, will, were, do, been, say, said, says”. For FOCUSED summarization, we modify this algorithm merely by running the sentence selection algorithm on only those sentences in the input that are relevent to the user query. In some previous DUC evaluations, relevant sentences are explicitly marked by annotators and given to systems. In our version here, a sentence in the input is considered relevant if it contains at least one word from the user query.

For evaluation we use ROUGE (Lin, 2004) SU4 recall metric[1], which was among the official automatic evaluation metrics for DUC.

4 Results

The results are shown in Table 1. The focused summarizer using LLR(CQ) is the best, and it *significantly* outperforms the focused summarizer based on frequency. Also, LLR (using log-likelihood ratio to assign weights to *all* words) perfroms significantly worse than LLR(C). We can observe some trends even from the results for which there is no significance. Both LLR and LLR(C) are sensitive to the introduction of topic relevance, producing somewhat better summaries in the FOCUSED scenario

[1] -n 2 -x -m -2 4 -u -c 95 -r 1000 -f A -p 0.5 -t 0 -d

compared to the GENERIC scenario. This is not the case for the frequency summarizer, where using only the relevant sentences has a negative impact.

4.1 Focused summarization: do we need query expansion?

In the FOCUSED condition there was little (for LLR weighting) or no (for frequency) improvement over GENERIC. One possible explanation for the lack of clear improvement in the FOCUSED setting is that there are not enough relevant sentences, making it impossible to get stable estimates of word importance. Alternatively, it could be the case that many of the sentences are relevant, so estimates from the relevant portion of the input are about the same as those from the entire input.

To distinguish between these two hypotheses, we conducted an oracle experiment. We modified the FOCUSED condition by expanding the topic words from the user query with *all* content words from any of the human-written summaries for the topic. This increases the number of relevant sentences for each topic. No automatic method for query expansion can be expected to give more accurate results, since the content of the human summaries is a direct indication of what information in the input was important and relevant and, moreover, the ROUGE evaluation metric is based on direct n-gram comparison with these human summaries.

Even under these conditions there was no significant improvement for the summarizers, each getting better by 0.002: the frequency summarizer gets R-SU4 of 0.12048 and the LLR(CQ) summarizer achieves R-SU4 of 0.12717.

These results seem to suggest that considering the content words in the user topic results in enough relevant sentences. Indeed, Table 2 shows the minimum, maximum and average percentage of relevant sentences in the input (containing at least one content words from the user the query), both as defined by the original query and by the oracle query expansion. It is clear from the table that, on average, over half of the input comprises sentences that are relevant to the user topic. Oracle query expansion makes the number of relevant sentences almost equivalent to the input size and it is thus not surprising that the corresponding results for content selection are nearly identical to the query independent runs of generic summaries for the entire input.

	Original query	Oracle query expansion
Min	13%	52%
Average	57%	86%
Max	82%	98%

Table 2: Percentage of relevant sentences (containing words from the user query) in the input. The oracle query expansion considers all content words form human summaries of the input as query words.

These numbers indictate that rather than finding ways for query expansion, it might instead be more important to find techniques for *constraining* the query, determining which parts of the input are directly related to the user questions. Such techniques have been described in the recent multi-strategy approach of Lacatusu et al. (2006) for example, where one of the strategies breaks down the user topic into smaller questions that are answered using robust question-answering techniques.

4.2 Why is log-likelihood ratio better than frequency?

Frequency and log-likelihood ratio weighting for content words produce similar results when applied to rank all words in the input, while the cut-off for topicality in LLR(C) does have a positive impact on content selection. A closer look at the two weighting schemes confirms that when cut-off is not used, similar weighting of content words is produced. The Spearman correlation coefficient between the weights for words assigned by the two schemes is on average 0.64. At the same time, it is likely that the weights of sentences are dominated by only the top most highly weighted words. In order to see to what extent the two schemes identify the same or different words as the most important ones, we computed the overlap between the 250 most highly weighted words according to LLR and frequency. The average overlap across the 50 sets was quite large, 70%.

To illustrate the degree of overlap, we list below are the most highly weighted words according to each weighting scheme for our sample topic concerning crimes across borders.

LLR *drug, cocaine, traffickers, cartel, police, crime, enforcement, u.s., smuggling, trafficking, arrested, government, seized, year, drugs, organised, heroin, criminal, cartels, last,* *official*, *country*, *law*, *border*, kilos, arrest, *more*, *mexican*, laundering, *officials*, *money*, *accounts*, *charges*, *authorities*, corruption, anti-drug, international, banks, operations, seizures, federal, italian, smugglers, dealers, narcotics, criminals, tons, most, planes, customs

Frequency *drug*, *cocaine*, *officials*, *police*, *more*, *last*, *government*, *year*, *cartel*, *traffickers*, *u.s.*, other, *drugs*, *enforcement*, *crime*, *money*, *country*, *arrested*, federal, most, now, *trafficking*, *seized*, *law*, years, new, *charges*, *smuggling*, being, *official*, *organised*, international, former, *authorities*, only, *criminal*, *border*, people, countries, state, world, trade, first, *mexican*, many, *accounts*, according, bank, *heroin*, *cartels*

It becomes clear that the advantage of likelihood ratio as a weighting scheme does not come from major differences in overall weights it assigns to words compared to frequency. It is the significance cut-off for the likelihood ratio that leads to noticeable improvement (see Table 1). When this weighting scheme is augmented by adding a score of 1 for content words that appear in the user topic, the summaries improve even further (LLR(CQ)). Half of the improvement can be attributed to the cut-off (LLR(C)), and the other half to focusing the summary using the information from the user query (LLR(CQ)). The advantage of likelihood ratio comes from its providing a principled criterion for deciding which words are truly descriptive of the input and which are not. Raw frequency provides no such cut-off.

5 Conclusions

In this paper we examined two weighting schemes for estimating word importance that have been successfully used in current systems but have not to-date been directly compared. Our analysis confirmed that log-likelihood ratio leads to better results, but not because it defines a more accurate assignment of importance than raw frequency. Rather, its power comes from the use of a known distribution that makes it possible to determine which words are truly descriptive of the input. Only when such words are viewed as equally important in defining the topic does this weighting scheme show improved performance. Using the significance cut-off and considering all words above it equally important is key.

Log-likelihood ratio summarizer is more sensitive to topicality or relevance and produces summaries that are better when it take the user request into account than when it does not. This is not the case for a summarizer based on frequency.

At the same time it is noteworthy that the generic summarizers perform about as well as their focused counterparts. This may be related to our discovery that on average 57% of the sentences in the document are relevant and that ideal query expansion leads to a situation in which almost all sentences in the input become relevant. These facts could be an unplanned side-effect from the way the test topics were produced: annotators might have been influenced by information in the input to be summarizied when defining their topic. Such observations also suggest that a competitive generic summarizer would be an appropriate baseline for the topic-focused task in future DUCs. In addition, including some irrelavant documents in the input might make the task more challenging and allow more room for advances in query expansion and other summary focusing techniques.

References

J. Conroy, J. Schlesinger, and D. O'Leary. 2006. Topic-focused multi-document summarization using an approximate oracle score. In *Proceedings of the COLING/ACL'06 (Poster Session)*.

F. Lacatusu, A. Hickl, K. Roberts, Y. Shi, J. Bensley, B. Rink, P. Wang, and L. Taylor. 2006. Lcc's gistexter at duc 2006: Multi-strategy multi-document summarization. In *Proceedings of DUC'06*.

C. Lin and E. Hovy. 2000. The automated acquisition of topic signatures for text summarization. In *Proceedings of COLING'00*.

C. Lin. 2004. Rouge: a package for automatic evaluation of summaries. In *Proceedings of the Workshop on Text Summarization Branches Out (WAS 2004)*.

H. P. Luhn. 1958. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2(2):159–165.

C. Manning and H. Schutze. 1999. *Foundations of Statistical Natural Language Processing*. MIT Press.

A. Nenkova, L. Vanderwende, and K. McKeown. 2006. A compositional context sensitive multi-document summarizer: Exploring the factors that influence summarization. In *Proceedings of ACM SIGIR'06*.

A. Nenkova. 2005. Automatic text summarization of newswire: lessons learned from the document understanding conference. In *Proceedings of AAAI'05*.

L. Vanderwende, H. Suzuki, and C. Brockett. 2006. Microsoft research at duc 2006: Task-focused summarization with sentence simplification and lexical expansion. In *Proceedings of DUC'06*.

Expanding Indonesian-Japanese Small Translation Dictionary Using a Pivot Language

Masatoshi Tsuchiya[†] **Ayu Purwarianti**[‡] **Toshiyuki Wakita**[‡] **Seiichi Nakagawa**[‡]
[†]Information and Media Center / [‡]Department of Information and Computer Sciences,
Toyohashi University of Technology
tsuchiya@imc.tut.ac.jp, {wakita,ayu,nakagawa}@slp.ics.tut.ac.jp

Abstract

We propose a novel method to expand a small existing translation dictionary to a large translation dictionary using a pivot language. Our method depends on the assumption that it is possible to find a pivot language for a given language pair on condition that there are both a large translation dictionary from the source language to the pivot language, and a large translation dictionary from the pivot language to the destination language. Experiments that expands the Indonesian-Japanese dictionary using the English language as a pivot language shows that the proposed method can improve performance of a real CLIR system.

1 Introduction

Rich cross lingual resources including large translation dictionaries are necessary in order to realize working cross-lingual NLP applications. However, it is infeasible to build such resources for all language pairs, because there are many languages in the world. Actually, while rich resources are available for several popular language pairs like the English language and the Japanese language, poor resources are only available for rest unfamiliar language pairs.

In order to resolve this situation, automatic construction of translation dictionary is effective, but it is quite difficult as widely known. We, therefore, concentrate on the task of expanding a small existing translation dictionary instead of it. Let us consider three dictionaries: a small *seed dictionary* which consists of headwords in the source language and their translations in the destination language, a large *source-pivot dictionary* which consists of headwords in the source language and their translations in the pivot language, and a large *pivot-destination dictionary* which consists of headwords in the pivot language and their translations in the destination language. When these three dictionaries are given, expanding the seed dictionary is to translate words in the source language that meets two conditions: (1) they are not contained in the seed dictionary, and (2) they can be translated to the destination language transitively referring both the source-pivot dictionary and the pivot-destination dictionary.

Obviously, this task depends on two assumptions: (a) the existence of the small seed dictionary, and (b) the existence of the pivot language which meets the condition that there are both a large source-pivot dictionary and a large pivot-destination dictionary. Because of the first assumption, it is true that this task cannot be applied to a brand-new language pair. However, the number of such brand-new language pairs are decreasing while machine-readable language resources are increasing. Moreover, The second assumption is valid for many language pairs, when supposing the English language as a pivot. From these point of view, we think that the expansion task is more promising, although it depends more assumptions than the construction task.

There are two different points among the expansion task and the construction task. Previous researches of the construction task can be classified into two groups. The first group consists of researches to construct a new translation dictionary for a fresh language pair from existing translation dictionaries or other language resources (Tanaka and Umemura, 1994). In the first group, information of the seed dictionary are not counted in them unlike the expansion task, because it is assumed that there is no seed dictionary for such fresh language pairs. The second group consists of researches to translate

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 197–200,
Prague, June 2007. ©2007 Association for Computational Linguistics

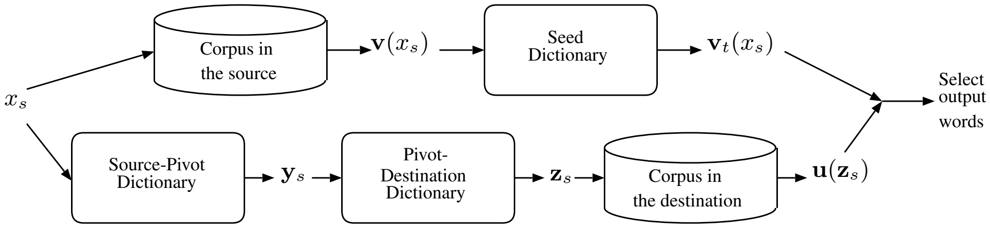


Figure 1: Translation Procedure

novel words using both a large existing translation dictionary and other linguistic resources like huge parallel corpora (Tonoike et al., 2005). Because almost of novel words are nouns, these researches focus into the task of translating nouns. In the expansion task, however, it is necessary to translate verbs and adjectives as well as nouns, because a seed dictionary will be so small that only basic words will be contained in it if the target language pair is unfamiliar. We will discuss about this topic in Section 3.2.

The remainder of this paper is organised as follows: Section 2 describes the method to expand a small seed dictionary. The experiments presented in Section 3 shows that the proposed method can improve performance of a real CLIR system. This paper ends with concluding remarks in Section 4.

2 Method of Expanding Seed Dictionary

The proposed method roughly consists of two steps shown in Figure 1. The first step is to generate a co-occurrence vector on the destination language corresponding to an input word, using both the seed dictionary and a monolingual corpus in the source language. The second step is to list translation candidates up, referring both the source-pivot dictionary and the pivot-destination dictionary, and to calculate their co-occurrence vectors based on a monolingual corpus in the destination.

The seed dictionary is used to convert a co-occurrence vector in the source language into a vector in the destination language. In this paper, $f(w_i, w_j)$ represents a co-occurrence frequency of a word w_i and a word w_j for all languages. A co-occurrence vector $\mathbf{v}(x_s)$ of a word x_s in the source is:

$$\mathbf{v}(x_s) = (f(x_s, x_1), \ldots, f(x_s, x_n)), \quad (1)$$

where $x_i (i = 1, 2, \ldots, n)$ is a headword of the seed dictionary D. A co-occurrence vector $\mathbf{v}(x_s)$, whose each element is corresponding to a word in the source, is converted into a vector $\mathbf{v}_t(x_s)$, whose each element is corresponding to a word in the destination, referring the dictionary D:

$$\mathbf{v}_t(x_s) = (f_t(x_s, z_1), \ldots, f_t(x_s, z_m)), \quad (2)$$

where $z_j (j = 1, 2, \ldots, m)$ is a translation word which appears in the dictionary D. The function $f_t(x_s, z_k)$, which assigns a co-occurrence degree between a word x_s and a word z_j in the destination based on a co-occurrence vector of a word x_s in the source, is defined as follows:

$$f_t(x_s, z_j) = \sum_{i=1}^{n} f(x_s, x_i) \cdot \delta(x_i, z_j). \quad (3)$$

where $\delta(x_i, z_j)$ is equal to one when a word z_j is included in a translation word set $D(x_i)$, which consists of translation words of a word x_i, and zero otherwise.

A set of description sentences $\mathbf{Y}_s$ in the pivot are obtained referring the source-pivot dictionary for a word x_s. After that, a description sentence $\mathbf{y}_s \in \mathbf{Y}_s$ in the pivot is converted to a set of description sentences $\mathbf{Z}_s$ in the destination referring the pivot-destination dictionary. A co-occurrence vector against a candidate description sentence $\mathbf{z}_s = z_s^1 z_s^2 \cdots z_s^l$, which is an instance of $\mathbf{Z}_s$, is calculated by this equation:

$$\mathbf{u}(\mathbf{z}_s) = \left(\sum_{k=1}^{l} f(z_s^k, z_1), \ldots, \sum_{k=1}^{l} f(z_s^k, z_m) \right) \quad (4)$$

Finally, the candidate $\mathbf{z}_s$ which meets a certain condition is selected as an output. Two conditions are examined in this paper: (1) selecting top-n candidates from sorted ones according to each similarity score, and (2) selecting candidates whose similarity scores are greater than a certain threshold. In this paper, cosine distance $s(\mathbf{v}_t(x_s), \mathbf{u}(\mathbf{z}_s))$ between a vector based on an input word x_s and a vector based on a candidate $\mathbf{z}_s$ is used as the similarity score between them.

3 Experiments

In this section, we present the experiments of the proposed method that the Indonesian language, the English language and the Japanese language are adopted as the source language, the pivot language and the destination language respectively.

3.1 Experimental Data

The proposed method depends on three translation dictionaries and two monolingual corpora as described in Section 2.

Mainichi Newspaper Corpus (1993–1995), which contains 3.5M sentences consist of 140M words, is used as the Japanese corpus. When measuring similarity between words using co-occurrence vectors, it is common that a corpus in the source language for the similar domain to one of the corpus in the source language is more suitable than one for a different domain. Unfortunately, because we could not find such corpus, the articles which were downloaded from the Indonesian Newspaper WEB sites[1] are used as the Indonesian corpus. It contains 1.3M sentences, which are tokenized into 10M words.

An online Indonesian-Japanese dictionary[2] contains 10,172 headwords, however, only 6,577 headwords of them appear in the Indonesian corpus. We divide them into two sets: the first set which consists of 6,077 entries is used as the seed dictionary, and the second set which consists of 500 entries is used to evaluate translation performance. Moreover, an online Indonesian-English dictionary[3], and an English-Japanese dictionary(Michibata, 2002) are also used as the source-pivot dictionary and the pivot-destination dictionary.

3.2 Evaluation of Translation Performance

As described in Section 2, two conditions of selecting output words among candidates are examined. Table 1 shows their performances and the baseline, that is the translation performance when all candidates are selected as output words. It is revealed that the condition of selecting top-n candidates outperforms the another condition and the baseline. The maximum $F_{\beta=1}$ value of 52.5% is achieved when selecting top-3 candidates as output words.

Table 2 shows that the lexical distribution of headwords contained in the seed dictionary are quite similar to the lexical distribution of headwords contained in the source-pivot dictionary. This observation means that it is necessary to translate verbs and adjectives as well as nouns, when expanding this seed dictionary. Table 3 shows translation performances against nouns, verbs and adjectives, when selecting top-3 candidates as output words. The proposed method can be regarded likely because it is effective to verbs and adjectives as well as to nouns, whereas the baseline precision of verbs is considerably lower than the others.

3.3 CLIR Performance Improved by Expanded Dictionary

In this section, performance impact is presented when the dictionary expanded by the proposed method is adopted to the real CLIR system proposed in (Purwarianti et al., 2007).

NTCIR3 Web Retrieval Task(Eguchi et al., 2003) provides the evaluation dataset and defines the evaluation metric. The evaluation metric consists of four MAP values: PC, PL, RC and RL. They are corresponding to assessment types respectively. The dataset consists 100GB Japanese WEB documents and 47 queries of Japanese topics. The Indonesian queries, which are manually translated from them, are used as inputs of the experiment systems. The number of unique words which occur in the queries is 301, and the number of unique words which are not contained in the Indonesian-Japanese dictionary is 106 (35%). It is reduced to 78 (26%), while the existing dictionary that contains 10,172 entries is expanded to the dictionary containing 20,457 entries with the proposed method.

Table 4 shows the MAP values achieved by both the baseline systems using the existing dictionary and ones using the expanded dictionary. The former three systems use existing dictionaries, and the latter three systems use the expanded one. The 3rd system translates keywords transitively using both

[1] `http://www.kompas.com/`, `http://www.tempointeraktif.com/`

[2] `http://m1.ryu.titech.ac.jp/~indonesia/todai/dokumen/kamusjpina.pdf`

[3] `http://nlp.aia.bppt.go.id/kebi`

Table 1: Comparison between Conditions of Selecting Output Words

	Selecting top-n candidates					Selecting plausible candidates				Baseline
	$n = 1$	$n = 2$	$n = 3$	$n = 5$	$n = 10$	$x = 0.1$	$x = 0.16$	$x = 0.2$	$x = 0.3$	
Prec.	55.4%	49.9%	46.2%	40.0%	32.2%	20.8%	23.6%	25.8%	33.0%	18.9%
Rec.	40.9%	52.6%	60.7%	67.4%	74.8%	65.3%	50.1%	40.0%	16.9%	82.5%
$F_{\beta=1}$	47.1%	51.2%	**52.5%**	50.2%	45.0%	31.6%	32.1%	31.4%	22.4%	30.8%

Table 2: Lexical Classification of Headwords

	Indonesian-Japanese	Indonesian-English
# of nouns	4085 (57.4%)	15718 (53.5%)
# of verbs	1910 (26.8%)	9600 (32.7%)
# of adjectives	795 (11.2%)	3390 (11.5%)
# of other words	330 (4.6%)	682 (2.3%)
Total	7120 (100%)	29390 (100%)

Table 3: Performance for Nouns, Verbs and Adjectives

	Noun		Verb		Adjective	
	$n = 3$	Baseline	$n = 3$	Baseline	$n = 3$	Baseline
Prec.	49.1%	21.8%	41.0%	14.7%	46.9%	26.7%
Rec.	65.6%	80.6%	52.3%	84.1%	59.4%	88.4%
$F_{\beta=1}$	56.2%	34.3%	46.0%	25.0%	52.4%	41.0%

Table 4: CLIR Performance

	PC	PL	RC	RL
(1) Existing Indonesian-Japanese dictionary	0.044	0.044	0.037	0.037
(2) Existing Indonesian-Japanese dictionary and Japanese proper name dictionary	0.054	0.052	0.047	0.045
(3) Indonesian-English-Japanese transitive translation with statistic filtering	0.078	0.072	0.055	0.053
(4) Expanded Indonesian-Japanese dictionary	0.061	0.059	0.046	0.046
(5) Expanded Indonesian-Japanese dictionary with Japanese proper name dictionary	0.066	0.063	0.049	0.049
(6) Expanded Indonesian-Japanese dictionary with Japanese proper name dictionary and statistic filtering	0.074	0.072	0.059	0.058

the source-pivot dictionary and the pivot-destination dictionary, and the others translate keywords using either the existing source-destination dictionary or the expanded one. The 3rd system and the 6th system try to eliminate unnecessary translations based statistic measures calculated from retrieved documents. These measures are effective as shown in (Purwarianti et al., 2007), but, consume a high run-time computational cost to reduce enormous translation candidates statistically. It is revealed that CLIR systems using the expanded dictionary outperform ones using the existing dictionary without statistic filtering. And more, it shows that ones using the expanded dictionary without statistic filtering achieve near performance to the 3rd system without paying a high run-time computational cost. Once it is paid, the 6th system achieves almost same score of the 3rd system. These observation leads that we can conclude that our proposed method to expand dictionary is valuable to a real CLIR system.

4 Concluding Remarks

In this paper, a novel method of expanding a small existing translation dictionary to a large translation dictionary using a pivot language is proposed. Our method uses information obtained from a small existing translation dictionary from the source language to the destination language effectively. Experiments that expands the Indonesian-Japanese dictionary using the English language as a pivot language shows that the proposed method can improve performance of a real CLIR system.

References

Koji Eguchi, Keizo Oyama, Emi Ishida, Noriko Kando, , and Kazuko Kuriyama. 2003. Overview of the web retrieval task at the third NTCIR workshop. In *Proceedings of the Third NTCIR Workshop on research in Information Retrieval, Automatic Text Summarization and Question Answering*.

Hideki Michibata, editor. 2002. *Eijiro*. ALC, 3. (in Japanese).

Ayu Purwarianti, Masatoshi Tsuchiya, and Seiichi Nakagawa. 2007. Indonesian-Japanese transitive translation using English for CLIR. *Journal of Natural Language Processing*, 14(2), Apr.

Kumiko Tanaka and Kyoji Umemura. 1994. Construction of a bilingual dictionary intermediated by a third language. In *Proceedings of the 15th International Conference on Computational Linguistics*.

Masatugu Tonoike, Mitsuhiro Kida, Toshihiro Takagi, Yasuhiro Sasaki, Takehito Utsuro, and Satoshi Sato. 2005. Translation estimation for technical terms using corpus collected from the web. In *Proceedings of the Pacific Association for Computational Linguistics*, pages 325–331, August.

Shallow Dependency Labeling

Manfred Klenner
Institute of Computational Linguistics
University of Zurich
klenner@cl.unizh.ch

Abstract

We present a formalization of dependency labeling with Integer Linear Programming. We focus on the integration of subcategorization into the decision making process, where the various subcategorization frames of a verb compete with each other. A maximum entropy model provides the weights for ILP optimization.

1 Introduction

Machine learning classifiers are widely used, although they lack one crucial model property: they can't adhere to prescriptive knowledge. Take grammatical role (GR) labeling, which is a kind of (shallow) dependency labeling, as an example: chunk-verb-pairs are classified according to a GR (cf. (Buchholz, 1999)). The trials are independent of each other, thus, local decisions are taken such that e.g. a unique GR of a verb might (erroneously) get multiply instantiated etc. Moreover, if there are alternative subcategorization frames of a verb, they must not be confused by mixing up GR from different frames to a non-existent one. Often, a subsequent filter is used to repair such inconsistent solutions. But usually there are alternative solutions, so the demand for an optimal repair arises.

We apply the optimization method Integer Linear Programming (ILP) to (shallow) dependency labeling in order to generate a globally optimized consistent dependency labeling for a given sentence. A maximum entropy classifier, trained on vectors with morphological, syntactic and positional information automatically derived from the TIGER treebank (German), supplies probability vectors that are used as weights in the optimization process. Thus, the probabilities of the classifier do not any longer provide (as usually) the solution (i.e. by picking out the most probable candidate), but count as probabilistic suggestions to a - globally consistent - solution. More formally, the dependency labeling problem is: given a sentence with (i) verbs, $\mathcal{VB}$, (ii) NP and PP chunks[1], $\mathcal{CH}$, label all pairs $(\mathcal{VB} \cup \mathcal{CH}) \times (\mathcal{VB} \cup \mathcal{CH})$ with a dependency relation (including a class for the null assignment) such that all chunks get attached and for each verb exactly one subcategorization frame is instantiated.

2 Integer Linear Programming

Integer Linear Programming is the name of a class of constraint satisfaction algorithms which are restricted to a numerical representation of the problem to be solved. The objective is to optimize (e.g. maximize) a linear equation called the objective function (a) in Fig. 1) given a set of constraints (b) in Fig. 1):

$$a)\ \max : f(X_1, \ldots, X_n) := y_1 X_1 + \ldots + y_n X_n$$

$$b)\ a_{i1}X_1 + a_{i2}X_2 + \ldots + a_{in}X_n \begin{pmatrix} \leq \\ = \\ \geq \end{pmatrix} b_i,$$

Figure 1: ILP Specification

where, $i = 1, \ldots, m$ and $X_1 \ldots X_n$ are variables, $y_1 \ldots y_n$, b_i and $a_{i1} \ldots a_{in}$ are constants.

For dependency labeling we have: X_n are binary class variables that indicate the (non-) assignment of a chunk c to a dependency relation G of a subcat frame f of a verb v. Thus, three indices are needed: G_{fvc}. If such an indicator variable G_{fvc} is set to 1 in the course of the maximization task, then the dependency label G between these chunks is said to hold, otherwise ($G_{fvc} = 0$) it doesn't hold. $y_1 \ldots y_n$ from Fig.1 are interpreted as weights that represent the impact of an assignment.

3 Dependency Labeling with ILP

Given the chunks $\mathcal{CH}^+$ (NP, PP and verbs) of a sentence, each pair $\mathcal{CH}^+ \times \mathcal{CH}^+$ is formed. It can

[1] Note that we use base chunks instead of heads.

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 201–204,
Prague, June 2007. ©2007 Association for Computational Linguistics

$$\mathcal{M} = \sum_{i}^{|CH|} \sum_{j\ (i \neq j)}^{|CH|} \omega_{\mathcal{T}_{ij}} * \mathcal{T}_{ij} \quad (1)$$

$$\mathcal{A} = \sum_{i}^{|VB|} \sum_{j}^{|PP|} \omega_{\mathcal{J}_{ij}} * \mathcal{J}_{ij} \quad (2)$$

$$\mathcal{V} = \sum_{c}^{|CH^{+}|} \sum_{v}^{|VB|} \sum_{\langle G,f\rangle \in R_v} w_{G_{fvc}} * G_{fvc} \quad (3)$$

$$\mathcal{U} = \sum_{i}^{|VB|} \sum_{j(i \neq j)}^{|VB|} \omega_{U_{ij}} * U_{ij} \quad (4)$$

$$\max : \mathcal{M} + \mathcal{A} + \mathcal{V} + \mathcal{U} \quad (5)$$

Figure 2: Objective Function

stand in one of eight dependency relations, including a pseudo relation representing the null class. We consider the most important dependency labels: subject ($\mathcal{S}$), direct object ($\mathcal{D}$), indirect object ($\mathcal{I}$), clausal complement ($\mathcal{C}$), prepositional complement ($\mathcal{P}$), attributive (NP or PP) attachment ($\mathcal{T}$) and adjunct ($\mathcal{J}$). Although coarse-grained, this set allows us to capture all functional dependencies and to construct a dependency tree for every sentence in the corpus[2]. Technically, indicator variables are used to represent attachment decisions. Together with a weight, they form the addend of the objective function. In the case of attributive modifiers or adjuncts (the non-governable labels), the indicator variables correspond to triples. There are two labels of this type: $\mathcal{T}_{ij}$ represents that chunk j modifies chunk i and $\mathcal{J}_{ij}$ represents that chunk j is in an adjunct relation to chunk i. $\mathcal{M}$ and $\mathcal{A}$ are defined as the weighted sum of such pairs (cf. Eq. 1 and Eq 2. from Fig. 2), the weights (e.g. $\omega_{\mathcal{T}_{ij}}$) stem from the statistical model.

For subcategorized labels, we have quadruples, consisting of a label name G, a frame index f, a verb v and a chunk c (also verb chunks are allowed as a c): G_{fvc}. We define $\mathcal{V}$ to be the weighted sum of all label instantiations of all verbs (and their subcat frames), see Eq. 3 in Fig. 2. The subscript R_v is a list of pairs, where each pair consists of a label and a subcat frame index. This way, R_v represents all subcat frames of a verb v. For example, R of “to believe” could be: $\{\langle\mathcal{S},1\rangle, \langle\mathcal{D},1\rangle, \langle\mathcal{S},2\rangle, \langle\mathcal{C},2\rangle, \langle\mathcal{S},3\rangle, \langle\mathcal{I},3\rangle\}$. There are three frames, the first one requires a $\mathcal{S}$ and a $\mathcal{D}$.

Consider the sentence “He believes these stories”. We have VB=$\{$believes$\}$ and CH^{+} = $\{$He, believes, stories$\}$. Assume R_1 to be the R of “to believe” as defined above. Then, e.g. $S_{213} = 1$ represents the assignment of “stories” as the filler of the subject relation S of the second subcat frame of “believes”.

To get a dependency tree, every chunk must find a head (chunk), except the root verb. We define a root verb j as a verb that stands in the relation $\mathcal{U}_{ij}$ to all other verbs i. $\mathcal{U}$ (cf. Eq.4 from Fig.2) is the weighted sum of all null assignment decisions. It is part of the maximization task and thus has an impact (a weight). The objective function is defined as the sum of equations 1 to 4 (Eq.5 from Fig.2).

So far, our formalization was devoted to the maximization task, i.e. which chunks are in a dependency relation, what is the label and what is the impact. Without any further (co-occurrence) restrictions, every pair of chunks would get related with every label. In order to assure a valid linguistic model, constraints have to be formulated.

4 Basic Global Constraints

Every chunk j from CH ($\neq CH^{+}$) must find a head, that is, be bound either as an attribute, adjunct or a verb complement. This requires all indicator variables with j as the dependent (second index) to sum up to exactly 1.

$$\sum_{c}^{|CH|} \mathcal{T}_{cj} + \sum_{i}^{|VB|} \mathcal{J}_{ij} + \sum_{v}^{|VB|} \sum_{\langle G,f\rangle \in R_v} G_{fvj} = 1, \quad (6)$$

$$\forall j : \ 0 < j \leq |CH|$$

A verb is attached to any other verb either as a clausal object $\mathcal{C}$ (of some verb frame f) or as $\mathcal{U}$ (null class) indicating that there is no dependency relation between them.

$$\mathcal{U}_{ij} + \sum_{\langle\mathcal{C},f\rangle \in R_i} \mathcal{C}_{fij} = 1, \ \forall i,j (i \neq j) : \ 0 < i,j \leq |VB| (7)$$

[2] Note that we are not interested in dependencies beyond the (base) chunk level

This does not exclude that a verb gets attached to several verbs as a $\mathcal{C}$. We capture this by constraint 8:

$$\sum_{i}^{|VB|} \sum_{\langle \mathcal{C},f\rangle \in R_i} \mathcal{C}_{fij} \leq 1,\ \ \forall j:\ 0 < j \leq |VB| \qquad (8)$$

Another (complementary) constraint is that a dependency label G of a verb must have at most one filler. We first introduce a indicator variable G_{fv}:

$$G_{fv} = \sum_{c}^{|CH^{+}|} G_{fvc} \qquad (9)$$

In order to serve as an indicator of whether a label G (of a frame f of a verb v) is active or inactive, we restrict G_{fv} to be at most 1:

$$G_{fv} \leq 1, \forall v, f, G : 0 < v \leq |VB| \wedge \langle G, f \rangle \in R_v (10)$$

To illustrate this by the example previously given: the subject of the second verb frame of "to believe" is defined as $S_{21} = \mathcal{S}_{211} + \mathcal{S}_{213}$ (with $S_{21} \leq 1$). Either $\mathcal{S}_{211} = 1$ or $\mathcal{S}_{213} = 1$ or both are zero, but if one of them is set to one, then S_{21} = 1. Moreover, as we show in the next section, the selection of the label indicator variable of a frame enforces the frame to be selected as well[3].

5 Subcategorization as a Global Constraint

The problem with the selection among multiple subcat frames is to guarantee a valid distribution of chunks to verb frames. We don't want to have chunk c_1 be labeled according to verb frame f_1 and chunk c_2 according to verb frame f_2. Any valid attachment must be coherent (address one verb frame) and complete (select all of its labels).

We introduce an indicator variable F_{fv} with frame and verb indices. Since exactly one frame of a verb has to be active at the end, we restrict:

$$\sum_{f=1}^{NF_v} \mathcal{F}_{fv} = 1,\ \ \forall v:\ \ 0 < v \leq |VB| \qquad (11)$$

(NF_v is the number of subcat frames of verb v)

However, we would like to couple a verb's (v) frame (f) to the frame's label set and restrict it to be active (i.e. set to one) only if all of its labels are active. To achieve this, we require equivalence, namely that selecting any label of a frame is equivalent to selecting the frame. As defined in equation 10, a label is active, if the label indicator variable (G_{fv}) is set to one. Equivalence is represented by identity, we thus get (cf. constraint 12):

$$\mathcal{F}_{fv} = G_{fv},\ \forall v, f, G : 0 < v \leq |VB| \wedge \langle G, f \rangle \in R_v (12)$$

If any G_{fv} is set to one (zero), then F_{fv} is set to one (zero) and all other G_{fv} of the same subcat frame are forced to be one (completeness). Constraint 11 ensures that exactly one subcat frame F_{fv} can be active (coherence).

6 Maximum Entropy and ILP Weights

A maximum entropy approach was used to induce a probability model that serves as the basis for the ILP weights. The model was trained on the TIGER treebank (Brants et al., 2002) with feature vectors stemming from the following set of features: the part of speech tags of the two candidate chunks, the distance between them in chunks, the number of intervening verbs, the number of intervening punctuation marks, person, case and number features, the chunks, the direction of the dependency relation (left or right) and a passive/active voice flag.

The output of the maxent model is for each pair of chunks a probability vector, where each entry represents the probability that the two chunks are related by a particular label ($\mathcal{S}, \mathcal{D} \ldots$ including $\mathcal{U}$).

7 Empirical Results

A 80% training set (32,000 sentences) resulted in about 700,000 vectors, each vector representing either a proper dependency labeling of two chunks, or a null class pairing. The accuracy of the maximum entropy classifier was 87.46%. Since candidate pairs are generated with only a few restrictions, most pairings are null class labelings. They form the majority class and thus get a strong bias. If we evaluate the dependency labels, therefore, the results drop appreciably. The maxent precision then is 62.73% (recall is 85.76%, f-measure is 72.46 %).

Our first experiment was devoted to find out how good our ILP approach was given that the correct subcat frame was pre-selected by an oracle. Only the decision which pairs are labeled with which dependency label was left to ILP (also the selection and assignment of the non subcategorized labels).

[3] There are more constraints, e.g. that no two chunks can be attached to each other symmetrically (being chunk and modifier of each other at the same time). We won't introduce them here.

There are 8000 sentence with 36,509 labels in the test set; ILP retrieved 37,173; 31,680 were correct. Overall precision is 85.23%, recall is 86.77%, the f-measure is 85.99% (F$_{pres}$ in Fig. 3).

	F$_{pres}$			F$_{comp}$		
	Prec	Rec	F-Mea	Prec	Rec	F-Mea
$\mathcal{S}$	91.4	86.1	88.7	90.3	80.9	85.4
$\mathcal{D}$	90.4	83.3	86.7	81.4	73.3	77.2
$\mathcal{I}$	88.5	76.9	82.3	75.8	55.5	64.1
$\mathcal{P}$	79.3	73.7	76.4	77.8	40.9	55.6
$\mathcal{C}$	98.6	94.1	96.3	91.4	86.7	89.1
$\mathcal{J}$	76.7	75.6	76.1	74.5	72.3	73.4
$\mathcal{T}$	75.7	76.9	76.3	74.1	74.2	74.2

Figure 3: Pre-selected versus Competing Frames

The results of the governable labels ($\mathcal{S}$ down to $\mathcal{C}$) are good, except PP complements ($\mathcal{P}$) with a f-measure of 76.4%. The errors made with F_{pres}: the wrong chunks are deemed to stand in a dependency relation or the wrong label (e.g. $\mathcal{S}$ instead of $\mathcal{D}$) was chosen for an otherwise valid pair. This is not a problem of ILP, but one of the statistical model - the weights do not discriminate well. Improvements of the statistical model will push ILP's precision.

Clearly, performance drops if we remove the subcat frame oracle letting all subcat frames of a verb compete with each other (F$_{comp}$, Fig.3). How close can F$_{comp}$ come to the oracle setting F$_{pres}$. The overall precision of the F$_{comp}$ setting is 81.8%, recall is 85.8% and the f-measure is 83.7% (f-measure of F$_{pres}$ was 85.9%). This is not too far away.

We have also evaluated how good our model is at finding the correct subcat frame (as a whole). First some statistics: In the test set are 23 different subcat frames (types) with 16,137 occurrences (token). 15,239 out of these are cases where the underlying verb has more than one subcat frame (only here do we have a selection problem). The precision was 71.5%, i.e. the correct subcat frame was selected in 10,896 out of 15,239 cases.

8 Related Work

ILP has been applied to various NLP problems including semantic role labeling (Punyakanok et al., 2004), which is similar to dependency labeling: both can benefit from verb specific information. Actually, (Punyakanok et al., 2004) take into account to some extent verb specific information. They disallow argument types a verb does not "subcategorize for" by setting an occurrence constraint. However, they do not impose *co*-occurrence restrictions as we do (allowing for competing subcat frames).

None of the approaches to grammatical role labeling tries to scale up to dependency labeling. Moreover, they suffer from the problem of inconsistent classifier output (e.g. (Buchholz, 1999)). A comparison of the empirical results is difficult, since e.g. the number and type of grammatical/dependency relations differ (the same is true wrt. German dependency parsers, e.g (Foth et al., 2005)). However, our model seeks to integrate the (probabilistic) output of such systems and - in the best case - boosts the results, or at least turn it into a consistent solution.

9 Conclusion and Future Work

We have introduced a model for shallow dependency labeling where data-driven and theory-driven aspects are combined in a principled way. A classifier provides empirically justified weights, linguistic theory contributes well-motivated global restrictions, both are combined under the regiment of optimization. The empirical results of our approach are promising. However, we have made idealized assumptions (small inventory of dependency relations and treebank derived chunks) that clearly must be replaced by a realistic setting in our future work.

Acknowledgment. I would like to thank Markus Dreyer for fruitful ("long distance") discussions and the (steadily improved) maximum entropy models.

References

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius and George Smith. 2002. The TIGER Treebank. *Proc. of the Wshp. on Treebanks and Linguistic Theories Sozopol*.

Sabine Buchholz, Jorn Veenstra and Walter Daelemans. 1999. Cascaded Grammatical Relation Assignment. *EMNLP-VLC'99, the Joint SIGDAT Conference on Empirical Methods in NLP and Very Large Corpora*.

Kilian Foth, Wolfgang Menzel, and Ingo Schröder. Robust parsing with weighted constraints. *Natural Language Engineering, 11(1):1-25* 2005.

Vasin Punyakanok, Dan Roth, Wen-tau Yih, and Dave Zimak. 2004. Semantic Role Labeling via Integer Linear Programming Inference. *COLING '04*.

Minimally Lexicalized Dependency Parsing

Daisuke Kawahara and **Kiyotaka Uchimoto**
National Institute of Information and Communications Technology,
3-5 Hikaridai Seika-cho Soraku-gun, Kyoto, 619-0289, Japan
{dk, uchimoto}@nict.go.jp

Abstract

Dependency structures do not have the information of phrase categories in phrase structure grammar. Thus, dependency parsing relies heavily on the lexical information of words. This paper discusses our investigation into the effectiveness of lexicalization in dependency parsing. Specifically, by restricting the degree of lexicalization in the training phase of a parser, we examine the change in the accuracy of dependency relations. Experimental results indicate that minimal or low lexicalization is sufficient for parsing accuracy.

1 Introduction

In recent years, many accurate phrase-structure parsers have been developed (e.g., (Collins, 1999; Charniak, 2000)). Since one of the characteristics of these parsers is the use of lexical information in the tagged corpus, they are called "lexicalized parsers". Unlexicalized parsers, on the other hand, achieved accuracies almost equivalent to those of lexicalized parsers (Klein and Manning, 2003; Matsuzaki et al., 2005; Petrov et al., 2006). Accordingly, we can say that the state-of-the-art lexicalized parsers are mainly based on unlexical (grammatical) information due to the sparse data problem. Bikel also indicated that Collins' parser can use bilexical dependencies only 1.49% of the time; the rest of the time, it backs off to condition one word on just phrasal and part-of-speech categories (Bikel, 2004).

This paper describes our investigation into the effectiveness of lexicalization in dependency parsing instead of phrase-structure parsing. Usual dependency parsing cannot utilize phrase categories, and thus relies on word information like parts of speech and lexicalized words. Therefore, we want to know the performance of dependency parsers that have minimal or low lexicalization.

Dependency trees have been used in a variety of NLP applications, such as relation extraction (Culotta and Sorensen, 2004) and machine translation (Ding and Palmer, 2005). For such applications, a fast, efficient and accurate dependency parser is required to obtain dependency trees from a large corpus. From this point of view, minimally lexicalized parsers have advantages over fully lexicalized ones in parsing speed and memory consumption.

We examined the change in performance of dependency parsing by varying the degree of lexicalization. The degree of lexicalization is specified by giving a list of words to be lexicalized, which appear in a training corpus. For minimal lexicalization, we used a short list that consists of only high-frequency words, and for maximal lexicalization, the whole list was used. Consequently, minimally or low lexicalization is sufficient for dependency accuracy.

2 Related Work

Klein and Manning presented an unlexicalized PCFG parser that eliminated all the lexicalized parameters (Klein and Manning, 2003). They manually split category tags from a linguistic view. This corresponds to determining the degree of lexicalization by hand. Their parser achieved an F_1 of 85.7% for section 23 of the Penn Treebank. Matsuzaki et al. and Petrov et al. proposed an automatic approach to

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 205–208,
Prague, June 2007. ©2007 Association for Computational Linguistics

Dependency accuracy (DA) Proportions of words, except punctuation marks, that are assigned the correct heads.

Root accuracy (RA) Proportions of root words that are correctly detected.

Complete rate (CR) Proportions of sentences whose dependency structures are completely correct.

Table 1: Evaluation criteria.

	DA	RA	CR
(Yamada and Matsumoto, 2003)	90.3	91.6	38.4
(Nivre and Scholz, 2004)	87.3	84.3	30.4
(Isozaki et al., 2004)	91.2	95.7	40.7
(McDonald et al., 2005)	90.9	94.2	37.5
(McDonald and Pereira, 2006)	91.5	N/A	42.1
(Corston-Oliver et al., 2006)	90.8	93.7	37.6
Our Base Parser	90.9	92.6	39.2

Table 2: Comparison of parser performance.

splitting tags (Matsuzaki et al., 2005; Petrov et al., 2006). In particular, Petrov et al. reported an F_1 of 90.2%, which is equivalent to that of state-of-the-art lexicalized parsers.

Dependency parsing has been actively studied in recent years (Yamada and Matsumoto, 2003; Nivre and Scholz, 2004; Isozaki et al., 2004; McDonald et al., 2005; McDonald and Pereira, 2006; Corston-Oliver et al., 2006). For instance, Nivre and Scholz presented a deterministic dependency parser trained by memory-based learning (Nivre and Scholz, 2004). McDonald et al. proposed an online large-margin method for training dependency parsers (McDonald et al., 2005). All of them performed experiments using section 23 of the Penn Treebank. Table 2 summarizes their dependency accuracies based on three evaluation criteria shown in Table 1. These parsers believed in the generalization ability of machine learners and did not pay attention to the issue of lexicalization.

3 Minimally Lexicalized Dependency Parsing

We present a simple method for changing the degree of lexicalization in dependency parsing. This method restricts the use of lexicalized words, so it is the opposite to tag splitting in phrase-structure parsing. In the remainder of this section, we first describe a base dependency parser and then report experimental results.

3.1 Base Dependency Parser

We built a parser based on the deterministic algorithm of Nivre and Scholz (Nivre and Scholz, 2004) as a base dependency parser. We adopted this algorithm because of its linear-time complexity.

In the algorithm, parsing states are represented by triples $\langle S, I, A\rangle$, where S is the stack that keeps the words being under consideration, I is the list of remaining input words, and A is the list of determined dependencies. Given an input word sequence, W, the parser is first initialized to the triple $\langle nil, W, \phi\rangle^1$. The parser estimates a dependency relation between two words (the top elements of stacks S and I). The algorithm iterates until the list I is empty. There are four possible operations for a parsing state (where t is the word on top of S, n is the next input word in I, and w is any word):

Left In a state $\langle t|S, n|I, A\rangle$, if there is no dependency relation $(t \rightarrow w)$ in A, add the new dependency relation $(t \rightarrow n)$ into A and pop S (remove t), giving the state $\langle S, n|I, A \cup (t \rightarrow n)\rangle$.

Right In a state $\langle t|S, n|I, A\rangle$, if there is no dependency relation $(n \rightarrow w)$ in A, add the new dependency relation $(n \rightarrow t)$ into A and push n onto S, giving the state $\langle n|t|S, I, A \cup (n \rightarrow t)\rangle$.

Reduce In a state $\langle t|S, I, A\rangle$, if there is a dependency relation $(t \rightarrow w)$ in A, pop S, giving the state $\langle S, I, A\rangle$.

Shift In a state $\langle S, n|I, A\rangle$, push n onto S, giving the state $\langle n|S, I, A\rangle$.

In this work, we used Support Vector Machines (SVMs) to predict the operation given a parsing state. Since SVMs are binary classifiers, we used the pair-wise method to extend them in order to classify our four-class task.

The features of a node are the word's lemma, the POS/chunk tag and the information of its child node(s). The lemma is obtained from the word form using a lemmatizer, except for numbers, which are replaced by "⟨num⟩". The context features are the two preceding nodes of node t (and t itself), the two succeeding nodes of node n (and n itself), and their

[1] We use "nil" to denote an empty list and $a|A$ to denote a list with head a and tail A.

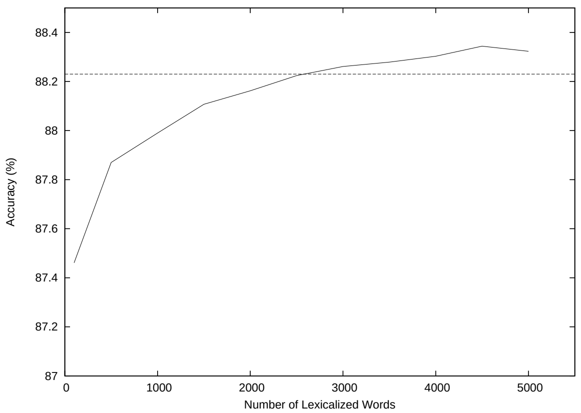


Figure 1: Dependency accuracies on the WSJ while changing the degree of lexicalization.

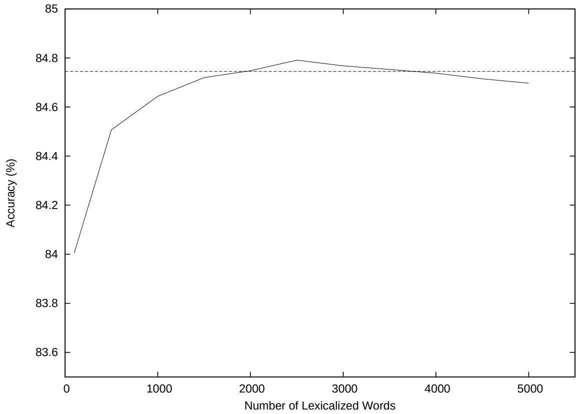


Figure 2: Dependency accuracies on the Brown Corpus while changing the degree of lexicalization.

child nodes (lemmas and POS tags). The distance between nodes n and t is also used as a feature.

We trained our models on sections 2-21 of the WSJ portion of the Penn Treebank. We used section 23 as the test set. Since the original treebank is based on phrase structure, we converted the treebank to dependencies using the head rules provided by Yamada [2]. During the training phase, we used intact POS and chunk tags[3]. During the testing phase, we used automatically assigned POS and chunk tags by Tsuruoka's tagger[4](Tsuruoka and Tsujii, 2005) and YamCha chunker[5](Kudo and Matsumoto, 2001). We used an SVMs package, TinySVM[6],and trained the SVMs classifiers using a third-order polynomial kernel. The other parameters are set to default.

The last row in Table 2 shows the accuracies of our base dependency parser.

3.2 Degree of Lexicalization vs. Performance

The degree of lexicalization is specified by giving a list of words to be lexicalized, which appear in a training corpus. For minimal lexicalization, we used a short list that consists of only high-frequency words, and for maximal lexicalization, the whole list was used.

To conduct the experiments efficiently, we trained our models using the first 10,000 sentences in sections 2-21 of the WSJ portion of the Penn Treebank. We used section 24, which is usually used as the development set, to measure the change in performance based on the degree of lexicalization.

We counted word (lemma) frequencies in the training corpus and made a word list in descending order of their frequencies. The resultant list consists of 13,729 words, and the most frequent word is "the", which occurs 13,252 times, as shown in Table 3. We define the degree of lexicalization as a threshold of the word list. If, for example, this threshold is set to 1,000, the top 1,000 most frequently occurring words are lexicalized.

We evaluated dependency accuracies while changing the threshold of lexicalization. Figure 1 shows the result. The dotted line (88.23%) represents the dependency accuracy of the maximal lexicalization, that is, using the whole word list. We can see that the decrease in accuracy is less than 1% at the minimal lexicalization (degree=100) and the accuracy of more than 3,000 degree slightly exceeds that of the maximal lexicalization. The best accuracy (88.34%) was achieved at 4,500 degree and significantly outperformed the accuracy (88.23%) of the maximal lexicalization (McNemar's test; $p = 0.017 < 0.05$). These results indicate that maximal lexicalization is not so effective for obtaining accurate dependency relations.

We also applied the same trained models to the Brown Corpus as an experiment of parser adaptation. We first split the Brown Corpus portion of

[2] http://www.jaist.ac.jp/˜h-yamada/

[3] In a preliminary experiment, we tried to use automatically assigned POS and chunk tags, but we did not detect significant difference in performance.

[4] http://www-tsujii.is.s.u-tokyo.ac.jp/˜tsuruoka/postagger/

[5] http://chasen.org/˜taku-ku/software/yamcha/

[6] http://chasen.org/˜taku-ku/software/TinySVM/

rank	word	freq.	rank	word	freq.
1	the	13,252	1,000	watch	29
2	,	12,858	⋮	⋮	⋮
⋮	⋮	⋮	2,000	healthvest	12
100	week	261	⋮	⋮	⋮
⋮	⋮	⋮	3,000	whoop	7
500	estate	64	⋮	⋮	⋮
⋮	⋮	⋮			

Table 3: Word list.

the Penn Treebank into training and testing parts in the same way as (Roark and Bacchiani, 2003). We further extracted 2,425 sentences at regular intervals from the training part and used them to measure the change in performance while varying the degree of lexicalization. Figure 2 shows the result. The dotted line (84.75%) represents the accuracy of maximal lexicalization. The resultant curve is similar to that of the WSJ experiment[7]. We can say that our claim is true even if the testing corpus is outside the domain.

3.3 Discussion

We have presented a minimally or lowly lexicalized dependency parser. Its dependency accuracy is close or almost equivalent to that of fully lexicalized parsers, despite the lexicalization restriction. Furthermore, the restriction reduces the time and space complexity. The minimally lexicalized parser (degree=100) took 12m46s to parse the WSJ development set and required 111 MB memory. These are 36% of time and 45% of memory reduction, compared to the fully lexicalized one.

The experimental results imply that training corpora are too small to demonstrate the full potential of lexicalization. We should consider unsupervised or semi-supervised ways to make lexicalized parsers more effective and accurate.

Acknowledgment

This research is partially supported by special coordination funds for promoting science and technology.

[7] In the experiment on the Brown Corpus, the difference between the best accuracy and the baseline was not significant.

References

Daniel M. Bikel. 2004. Intricacies of Collins' parsing model. *Computational Linguistics*, 30(4):479–511.

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proceedings of NAACL2000*, pages 132–139.

Michael Collins. 1999. *Head-Driven Statistical Models for Natural Language Parsing*. Ph.D. thesis, University of Pennsylvania.

Simon Corston-Oliver, Anthony Aue, Kevin Duh, and Eric Ringger. 2006. Multilingual dependency parsing using bayes point machines. In *Proceedings of HLT-NAACL2006*, pages 160–167.

Aron Culotta and Jeffrey Sorensen. 2004. Dependency tree kernels for relation extraction. In *Proceedings of ACL2004*, pages 423–429.

Yuan Ding and Martha Palmer. 2005. Machine translation using probabilistic synchronous dependency insertion grammars. In *Proceedings of ACL2005*, pages 541–548.

Hideki Isozaki, Hideto Kazawa, and Tsutomu Hirao. 2004. A deterministic word dependency analyzer enhanced with preference learning. In *Proceedings of COLING2004*, pages 275–281.

Dan Klein and Christopher D. Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of ACL2003*, pages 423–430.

Taku Kudo and Yuji Matsumoto. 2001. Chunking with support vector machines. In *Proceedings of NAACL2001*, pages 192–199.

Takuya Matsuzaki, Yusuke Miyao, and Jun'ichi Tsujii. 2005. Probabilistic CFG with latent annotations. In *Proceedings of ACL2005*, pages 75–82.

Ryan McDonald and Fernando Pereira. 2006. Online learning of approximate dependency parsing algorithms. In *Proceedings of EACL2006*, pages 81–88.

Ryan McDonald, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of ACL2005*, pages 91–98.

Joakim Nivre and Mario Scholz. 2004. Deterministic dependency parsing of English text. In *Proceedings of COLING2004*, pages 64–70.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of COLING-ACL2006*, pages 433–440.

Brian Roark and Michiel Bacchiani. 2003. Supervised and unsupervised PCFG adaptation to novel domains. In *Proceedings of HLT-NAACL2003*, pages 205–212.

Yoshimasa Tsuruoka and Jun'ichi Tsujii. 2005. Bidirectional inference with the easiest-first strategy for tagging sequence data. In *Proceedings of HLT-EMNLP2005*, pages 467–474.

Hiroyasu Yamada and Yuji Matsumoto. 2003. Statistical dependency analysis with support vector machines. In *Proceedings of IWPT2003*, pages 195–206.

HunPos – an open source trigram tagger

Péter Halácsy
Budapest U. of Technology
MOKK Media Research
H-1111 Budapest, Stoczek u 2
peter@halacsy.com

András Kornai
MetaCarta Inc.
350 Massachusetts Ave.
Cambridge MA 02139
andras@kornai.com

Csaba Oravecz
Hungarian Academy of Sciences
Institute of Linguistics
H-1068 Budapest, Benczur u. 33.
oravecz@nytud.hu

Abstract

In the world of non-proprietary NLP software the standard, and perhaps the best, HMM-based POS tagger is TnT (Brants, 2000). We argue here that some of the criticism aimed at HMM performance on languages with rich morphology should more properly be directed at TnT's peculiar license, free but not open source, since it is those details of the implementation which are hidden from the user that hold the key for improved POS tagging across a wider variety of languages. We present HunPos[1], a free and open source (LGPL-licensed) alternative, which can be tuned by the user to fully utilize the potential of HMM architectures, offering performance comparable to more complex models, but preserving the ease and speed of the training and tagging process.

0 Introduction

Even without a formal survey it is clear that TnT (Brants, 2000) is used widely in research labs throughout the world: Google Scholar shows over 400 citations. For research purposes TnT is freely available, but only in executable form (closed source). Its greatest advantage is its speed, important both for a fast tuning cycle and when dealing with large corpora, especially when the POS tagger is but one component in a larger information retrieval, information extraction, or question answering system. Though taggers based on dependency networks (Toutanova et al., 2003), SVM (Giménez and Màrquez, 2003), MaxEnt (Ratnaparkhi, 1996), CRF (Smith et al., 2005), and other methods may reach slightly better results, their train/test cycle is orders of magnitude longer.

A ubiquitous problem in HMM tagging originates from the standard way of calculating lexical probabilities by means of a lexicon generated during training. In highly inflecting languages considerably more unseen words will be present in the test data than in more isolating languages, which largely accounts for the drop in the performance of n-gram taggers when moving away from English. To mitigate the effect one needs a morphological dictionary (Hajič et al., 2001) or a morphological analyzer (Hakkani-Tür et al., 2000), but if the implementation source is closed there is no handy way to incorporate morphological knowledge in the tagger.

The paper is structured as follows. In Section 1 we present our own system, HunPos, while in Section 2 we describe some of the implementation details of TnT that we believe influence the performance of a HMM based tagging system. We evaluate the system and compare it to TnT on a variety of tasks in Section 3. We don't necessarily consider HunPos to be significantly better than TnT, but we argue that we could reach better results, *and so could others coming after us*, because the system is open to explore all kinds of fine-tuning strategies. Some concluding remarks close the paper in Section 4.

[1] http://mokk.bme.hu/resources/hunpos/

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 209–212,
Prague, June 2007. ©2007 Association for Computational Linguistics

1 Main features of HunPos

HunPos has been implemented in OCaml, a high-level language which supports a succinct, well-maintainable coding style. OCaml has a high-performance native-code compiler (Doligez et al., 2004) that can produce a C library with the speed of a C/C++ implementation.

On the whole HunPos is a straightforward trigram system estimating the probabilities

$$\operatorname*{argmax}_{t_1 \ldots t_T} P(t_{T+1}|t_T) \prod_{i=1}^{T} P(t_i|t_{i-1}, t_{i-2}) P(w_i|t_{i-1}, t_i)$$

for a given sequence of words $w_1 \ldots w_T$ (the additional tags t_{-1}, t_0, and t_{T+1} are for sentence boundary markers). Notice that unlike traditional HMM models, we estimate emission/lexicon probabilities based on the current tag and the previous tag as well. As we shall see in the next Section, using tag bigrams to condition the emissions can lead to as much as 10% reduction in the error rate. (In fact, HunPos can handle a context window of any size, but on the limited training sets available to us increasing this parameter beyond 2 gives no further improvement.)

As for contextualized lexical probabilities, our extension is very similar to Banko and Moore (2004) who use $P(w_i|t_{i-1}, t_i, t_{i+1})$ lexical probabilities and found, on the Penn Treebank, that “incorporating more context into an HMM when estimating lexical probabilities improved accuracy from 95.87% to 96.59%”. One difficulty with their approach, noted by Banko and Moore (2004), is the treatment of unseen words: their method requires a full dictionary that lists what tags are possible for each word. To be sure, for isolating languages such information is generally available from machine readable dictionaries which are often large enough to make the out of vocabulary problem negligible. But in our situation this amounts to idealized morphological analyzers (MA) that have their stem list extended so as to have no OOV on the test set.

The strong side of TnT is its suffix guessing algorithm that is triggered by unseen words. From the training set TnT builds a trie from the endings of words appearing less than n times in the corpus, and memorizes the tag distribution for each suffix.[2] A clear advantage of this approach is the probabilistic weighting of each label, however, under default settings the algorithm proposes a lot more possible tags than a morphological analyzer would. To facilitate the use of MA, HunPos has hooks to work with a morphological analyzer (lexicon), which might still leave some OOV items. As we shall see in Section 3, the key issue is that for unseen words the HMM search space may be narrowed down to the alternatives proposed by this module, which not only speeds up search but also very significantly improves precision. That is, for unseen words the MA will generate the possible labels, to which the weights are assigned by the suffix guessing algorithm.

2 Inside TnT

Here we describe, following the lead of (Jurish, 2003), some non-trivial features of TnT sometimes only hinted at in the user guide, but clearly evident from its behavior on real and experimentally adjusted corpora. For the most part, these features are clever hacks, and it is unfortunate that neither Brants (2000) nor the standard HMM textbooks mention them, especially as they often yield more significant error reduction than the move from HMM to other architectures. Naturally, these features are also available in HunPos.

2.1 Cardinals

For the following regular expressions TnT learns the tag distribution of the training corpus separately to give more reliable estimates for open class items like numbers unseen during training:

```
ˆ[0-9]+$
ˆ[0-9]+\.$
ˆ[0-9.,:-]+[0-9]+$
ˆ[0-9]+[a-zA-Z]{1,3}$
```

(The regexps are only inferred – we haven’t attempted to trace the execution.) After this, at test time, if the word is not found in the lexicon (numerals are added to the lexicon like all other items) TnT checks whether the unseen word matches some of the regexps, and uses the distribution learned for this regexp to guess the tag.

[2] The parameter n cannot be externally set — it is documented as 10 but we believe it to be higher.

2.2 Upper- and lowercase

The case of individual words may carry relevant information for tagging, so it is well worth preserving the uppercase feature for items seen as such in training. For unseen words TnT builds two suffix tries: if the word begins with uppercase one trie is used, for lowercase words the other trie is applied. The undocumented trick is to try to lookup the word in sentence initial position from the training lexicon in its lowercase variant, which contributes noticeably to the better performance of the system.

3 Evaluation

English For the English evaluation we used the WSJ data from Penn Treebank II. We extracted sentences from the parse trees. We split data into training and test set in the standard way (Table 1).

Set	Sect'ns	Sent.	Tokens	Unseen
Train	0-18	38,219	912,344	0
Test	22-24	5,462	129,654	2.81%

Table 1: Data set splits used for English

As Table 2 shows HunPos achieves performance comparable to TnT for English. The increase in the emission order clearly improves this performance.

	seen	unseen	overall
TnT	96.77%	85.91%	96.46%
HunPos 1	96.76%	86.90%	96.49%
HunPos 2	96.88%	86.13%	**96.58%**

Table 2: WSJ tagging accuracy, HunPos with first and second order emission/lexicon probabilities

If we follow Banko and Moore (2004) and construct a full (no OOV) morphological lexicon from the tagged version of the test corpus, we obtain 96.95% precision where theirs was 96.59%. For words seen, precision improves by an entirely negligible 0.01%, but for unseen words it improves by 10%, from 86.13% to 98.82%. This surprising result arises from the fact that there are a plenty of unambiguous tokens (especially the proper names that are usually unseen) in the test corpus.

What this shows is not just that morphology matters (this is actually not that visible for English), but that the difference between systems can only be appreciated once the small (and scantily documented) tricks are factored out. The reason why Banko and Moore (2004) get less than HunPos is not because their system is inherently worse, but rather because it lacks the engineering hacks built into TnT and HunPos.

Hungarian We evaluated the different models by tenfold cross-validation on the Szeged Corpus (Csendes et al., 2004), with the relevant data in presented Table 3.

Set	Sent.	Tokens	Unseens	OOV
Train	63,075	1,044,914	0	N.A
Test	7,008	116,101	9.59%	5.64%

Table 3: Data set splits used for Hungarian.

Note that the proportion of unseen words, nearly 10%, is more than three times higher than in English. Most of these words were covered by the morphological analyzer (Trón et al., 2006) but still 28% of unseen words were only guessed. However, this is just 2.86% of the whole corpus, in the magnitude similar to English.

morph	lex order	seen	unseen	overall
no	1	98.34%	88.96%	97.27%
	2	98.58%	87.97%	97.40%
yes	1	98.32%	96.01%	98.03%
	2	98.56%	95.96%	**98.24%**

Table 4: Tagging accuracy for Hungarian of HunPos with and without morphological lexicon and with first and second order emission/lexicon probabilities.

On the same corpus TnT had 97.42% and Halácsy et al. (2006) reached 98.17% with a MaxEnt tagger that used the TnT output as a feature. HunPos gets as good performance *in one minute* as this MaxEnt model which took three hours to go through the train/test cycle.

4 Concluding remarks

Though there can be little doubt that the ruling system of bakeoffs actively encourages a degree of oneupmanship, our paper and our software are not offered in a competitive spirit. As we said at the out-set, we don't necessarily believe HunPos to be in any way better than TnT, and certainly the main ideas have been pioneered by DeRose (1988), Church (1988), and others long before this generation of HMM work. But to improve the results beyond what a basic HMM can achieve one needs to tune the system, and progress can only be made if the experiments are end to end replicable.

There is no doubt many other systems could be tweaked further and improve on our results – what matters is that anybody could now also tweak HunPos without any restriction to improve the state of the art. Such tweaking can bring surprising results, e.g. the conclusion, strongly supported by the results presented here, that HMM tagging is actually quite competitive with, and orders of magnitude faster than, the current generation of learning algorithms including SVM and MaxEnt. No matter how good TnT was to begin with, the closed source has hindered its progress to the point that inherently clumsier, but better tweakable algorithms could overtake HMMs, a situation that HunPos has now hopefully changed at least for languages with more complex morphologies.

Acknowledgement

We thank Thorsten Brants for TnT, and György Gyepesi for constant help and encouragement.

References

Michele Banko and Robert C. Moore. 2004. Part of speech tagging in context. In *COLING '04: Proceedings of the 20th international conference on Computational Linguistics*, page 556, Morristown, NJ, USA. Association for Computational Linguistics.

Thorsten Brants. 2000. TnT – a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference (ANLP-2000)*, Seattle, WA.

Kenneth Ward Church. 1988. A stochastic parts program and noun phrase parser for unrestricted text. In *Proceedings of the second conference on Applied natural language processing*, pages 136–143, Morristown, NJ, USA. Association for Computational Linguistics.

Dóra Csendes, János Csirik, and Tibor Gyimóthy. 2004. The Szeged Corpus: A POS tagged and syntactically annotated Hungarian natural language corpus. In Karel Pala Petr Sojka, Ivan Kopecek, editor, *Text, Speech and Dialogue: 7th International Conference, TSD*, pages 41–47.

Steven J. DeRose. 1988. Grammatical category disambiguation by statistical optimization. *Computational Linguistics*, 14:31–39.

Damien Doligez, Jacques Garrigue, Didier Rémy, and Jérôme Vouillon, 2004. *The Objective Caml system*. Institut National de Recherche en Informatique et en Automatique.

Jesús Giménez and Lluís Màrquez. 2003. Fast and accurate part-of-speech tagging: The svm approach revisited. In *Proceedings of RANLP*, pages 153–163.

Jan Hajič, Pavel Krbec, Karel Oliva, Pavel Květoň, and Vladimír Petkevič. 2001. Serial combination of rules and statistics: A case study in Czech tagging. In *Proceedings of the 39th Association of Computational Linguistics Conference*, pages 260–267, Toulouse, France.

Dilek Z. Hakkani-Tür, Kemal Oflazer, and Gökhan Tür. 2000. Statistical morphological disambiguation for agglutinative languages. In *Proceedings of the 18th conference on Computational linguistics*, pages 285–291, Saarbrücken, Germany.

Péter Halácsy, András Kornai, Csaba Oravecz, Viktor Trón, and Dániel Varga. 2006. Using a morphological analyzer in high precision POS tagging of Hungarian. In *Proceedings of LREC 2006*, pages 2245–2248.

Bryan Jurish. 2003. A hybrid approach to part-of-speech tagging. Technical report, Berlin-Brandenburgische Akademie der Wissenschaften.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In Karel Pala Petr Sojka, Ivan Kopecek, editor, *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142, University of Pennsylvania.

Noah A. Smith, David A. Smith, and Roy W. Tromble. 2005. Context-based morphological disambiguation with random fields. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Vancouver.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL*, pages 252–259.

Viktor Trón, Péter Halácsy, Péter Rebrus, András Rung, Péter Vajda, and Eszter Simon. 2006. Morphdb.hu: Hungarian lexical database and morphological grammar. In *Proceedings of LREC 2006*, pages 1670–1673.

Extending MARIE: an N-gram-based SMT decoder

Josep M. Crego
TALP Research Center
Universitat Politècnica de Catalunya
Barcelona, 08034
jmcrego@gps.tsc.upc.edu

José B. Mariño
TALP Research Center
Universitat Politècnica de Catalunya
Barcelona,08034
canton@gps.tsc.upc.edu

Abstract

In this paper we present several extensions of MARIE[1], a freely available N-gram-based statistical machine translation (SMT) decoder. The extensions mainly consist of the ability to accept and generate word graphs and the introduction of two new N-gram models in the log-linear combination of feature functions the decoder implements. Additionally, the decoder is enhanced with a caching strategy that reduces the number of N-gram calls improving the overall search efficiency. Experiments are carried out over the Eurpoean Parliament Spanish-English translation task.

1 Introduction

Research on SMT has been strongly boosted in the last few years, partially thanks to the relatively easy development of systems with enough competence as to achieve rather competitive results. In parallel, tools and techniques have grown in complexity, which makes it difficult to carry out state-of-the-art research without sharing some of this toolkits. Without aiming at being exhaustive, GIZA++[2], SRILM[3] and PHARAOH[4] are probably the best known examples.

We introduce the recent extensions made to an N-gram-based SMT decoder (Crego et al., 2005), which allowed us to tackle several translation issues (such as reordering, rescoring, modeling, etc.) successfully improving accuracy, as well as efficiency results.

As far as SMT can be seen as a double-sided problem (modeling and search), the decoder emerges as a key component, core module of any SMT system. Mainly, any technique aiming at dealing with a translation problem needs for a decoder extension to be implemented. Particularly, the reordering problem can be more efficiently (and accurate) addressed when tightly coupled with decoding. In general, the competence of a decoder to make use of the maximum of information in the global search is directly connected with the likeliness of successfully improving translations.

The paper is organized as follows. In Section 2 we and briefly review the previous work on decoding with special attention to N-gram-based decoding. Section 3 describes the extended log-linear combination of feature functions after introduced the two new models. Section 4 details the particularities of the input and output word graph extensions. Experiments are reported on section 5. Finally, conclusions are drawn in section 6.

2 Related Work

The decoding problem in SMT is expressed by the next maximization: $\arg\max_{t_1^I \in \tau} P(t_1^I | s_1^J)$, where s_1^J is the source sentence to translate and t_1^I is a possible translation of the set τ, which contains all the sentences of the language of t_1^I.

Given that the full search over the whole set of target language sentences is impracticable (τ is an infinite set), the translation sentence is usually built incrementally, composing partial translations of the source sentence, which are selected out of a limited number of translation candidates (translation units).

The first SMT decoders were **word-based**. Hence, working with translation candidates of single source words. Later appeared the **phrase-based** decoders, which use translation candidates composed of sequences of source and target words (outperforming the word-based decoders by introducing the word context). In the last few years **syntax-based** decoders have emerged aiming at dealing with pair of languages with different syntactical structures for which the word context introduced

[1] http://gps-tsc.upc.es/soft/soft/marie

[2] http://www.fjoch.com/GIZA++.html

[3] http://www.speech.sri.com/projects/srilm/

[4] http://www.isi.edu/publications/licensed-sw/pharaoh/

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 213–216,
Prague, June 2007.

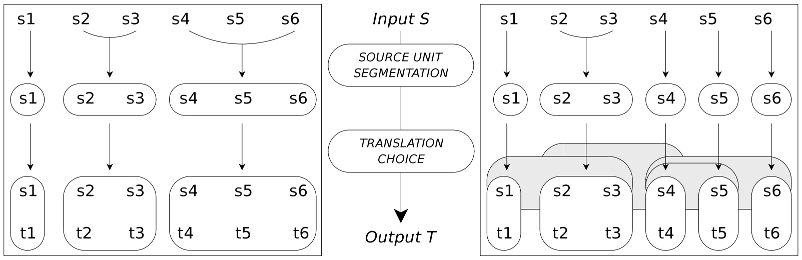


Figure 1: *Generative process. Phrase-based (left) and N-gram-based (right) approaches.*

in phrase-based decoders is not sufficient to cope with long reorderings.

Like standard phrase-based decoders, MARIE employs translation units composed of sequences of source and target words. In contrast, the translation context is differently taken into account. Whereas phrase-based decoders employ translation units uncontextualized, MARIE takes the translation unit context into account by estimating the translation model as a standard N-gram language model (N**-gram-based** decoder).

Figure 1 shows that both approaches follow the same generative process, but they differ on the structure of translation units. In the example, the units *’s1#t1’* and *’s2_s3#t2_t3’* of the N-gram-based approach are used considering that both appear sequentially. This fact can be understood as using a longer unit that includes both (longer units are drawn in grey).

MARIE follows the maximum entropy framework, where we can define a translation hypothesis t given a source sentence s, as the target sentence maximizing a log-linear combination of feature functions:

$$\hat{t}_1^I = \arg\max_{t_1^I} \left\{ \sum_{m=1}^{M} \lambda_m h_m(s_1^J, t_1^I) \right\} \qquad (1)$$

where λ_m corresponds to the weighting coefficients of the log-linear combination, and the feature functions $h_m(s,t)$ to a logarithmic scaling of the probabilities of each model. See (Mariño et al., 2006) for further details on the N-gram-based approach to SMT.

3 N-gram Feature Functions

Two language models (LM) are introduced in equation 1, aiming at helping the decoder to find the right translations. Both are estimated as standard N-gram LM.

3.1 Target-side N-gram LM

The first additional N-gram LM is destinated to be applied over the target sentence (tagged) words. Hence, as the original target LM (computed over raw words), it is also used to score the fluency of target sentences, but aiming at achieving generalization power through using a more generalized language (such as a language of Part-of-Speech tags) instead of the one composed of raw words. Part-Of-Speech tags have successfully been used in several previous experiments. however, any other tag can be applied.

Several sequences of target tags may apply to any given translation unit (which are passed to the decoder before it starts the search). For instance, regarding a translation unit with the english word ’*general*’ in its target side, if POS tags were used as target tagged tags, there would exist at least two different tag options: *noun* and *adjective*.

In the search, multiple hypotheses are generated concerning different target tagged sides (sequences of tags) of a single translation unit. Therefore, on the one side, the overall search is extended towards seeking the sequence of target tags that better fits the sequence of target raw words. On the other side, this extension is hurting the overall efficiency of the decoder as additional hypotheses appear in the search stacks while not additional translation hypotheses are being tested (only differently tagged).

This extended feature may be used toghether with a limitation of the number of target tagged hypotheses per translation unit. The use of a limited number of these hypotheses implies a balance between accuracy and efficiency.

3.2 Source-side N-gram LM

The second N-gram LM is applied over the input sentence tagged words. Obviously, this model only makes sense when reordering is applied over the source words in order to monotonize the source and target word order. In such a case, the tagged LM is learnt over the training set with reordered source words.

Hence, the new model is employed as a reordering model. It scores a given source-side reordering hypothesis according to the reorderings made in the training sentences (from which the tagged LM is estimated). As for the previous extension, source tagged words are used instead of raw words in order to achieve generalization power.

Additional hypotheses regarding the same translation unit are not generated in the search as all input sentences are uniquely tagged.

Figure 2 illustrates the use of a source POS-tagged N-

gram LM. The probability of the sequence ’*PRN VRB NAME ADJ*’ is greater than the probability of the sequence ’*PRN VRB ADJ NAME*’ for a model estimated over the training set with reordered source words (with english words following the spanish word order).

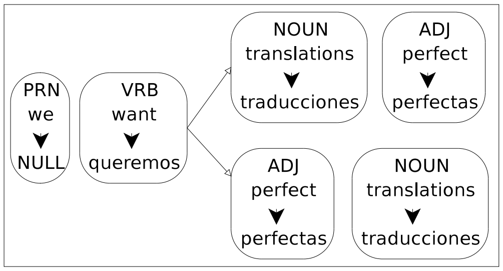


Figure 2: *Source POS-tagged N-gram LM.*

3.3 Caching N-grams

The use of several N-gram LM’s implies a reduction in efficiency in contrast to other models that can be implemented by means of a single lookup table (one access per probability call). The special characteristics of Ngram LM’s introduce additional memory access to account for backoff probabilities and lower Ngrams fallings.

Many N-gram calls are requested repeatedly, producing multiple calls of an entry. A simple strategy to reduce additional access consists of keeping a record (**cache**) for those Ngram entries already requested. A drawback for the use of a cache consists of the additional memory access derived of the cache maintenance (adding new and checking for existing entries).

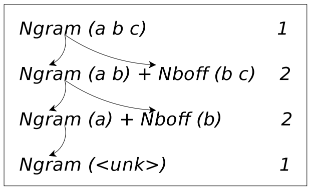


Figure 3: *Memory access derived of an N-gram call.*

Figure 3 illustrates this situation. The call for a 3-gram probability (requesting for the probability of the sequence of tokens ’*a b c*’) may need for up to 6 memory access, while under a phrase-based translation model the final probability would always be reached after the first memory access. The additional access in the N-gram-based approach are used to provide lower N-gram and backoff probabilities in those cases that upper N-gram probabilities do not exist.

4 Word Graphs

Word graphs are successfully used in SMT for several applications. Basically, with the objective of reducing the redundancy of N-best lists, which very often convey serious combinatorial explosion problems.

A word graph is here described as a directed acyclic graph $G = (V, E)$ with one root node $n_0 \in V$. Edges are labeled with tokens (words or translation units) and optionally with accumulated scores. We will use $(n_s(n_e\,''t''\,s))$, to denote an edge starting at node n_s and ending at node n_e, with token t and score s. The file format of word graphs coincides with the graph file format recognized by the CARMEL[5] finite state automata toolkit.

4.1 Input Graph

We can mainly find two applications for which word graphs are used as input of an SMT system: the recognition output of an automatic speech recognition (ASR) system; and a reordering graph, consisting of a subset of the whole word permutations of a given input sentence.

In our case we are using the input graph as a **reordering graph**. The decoder introduces reordering (distortion of source words order) by allowing only for the distortion encoded in the input graph. Though, the graph is only allowed to encode permutations of the input words. In other words, any path in the graph must start at node n_0, finish at node n_N (where n_N is a unique ending node) and cover all the input words (tokens t) in whatever order, without repetitions.

An additional feature function (distortion model) is introduced in the log-linear combination of equation 1:

$$p_{distortion}(u_k) \approx \prod_{i=k_1}^{k_I} p(n_i|n_{i-1}) \tag{2}$$

where u_k refers to the k^{th} partial translation unit covering the source positions $[k_1, ..., k_I]$. $p(n_i|n_{i-1})$ corresponds to the edge score s encoded in the edge $(n_s(n_e\,''t''\,s))$, where $n_i = n_e$ and $n_{i-1} = n_s$.

One of the decoding first steps consists of building (for each input sentence) the set of translation units to be used in the search. When the search is extended with reordering abilities the set must be also extended with those translation units that cover any sequence of input words following any of the word orders encoded in the input graph. The extension of the units set is specially relevant when translation units are built from the tranining set with reordered source words.

Given the example of figure 2, if the translation unit ’*translations perfect # traducciones perfectas*’ is available, the decoder should not discard it, as it provides a right translation. Notwithstanding that its source side does not follow the original word order of the input sentence.

4.2 Output Graph

The goal of using an output graph is to allow for further rescoring work. That is, to work with alternative transla-

[5] http://www.isi.edu/licensed-sw/carmel/

tions to the single 1-best. Therefore, our proposed output graph has some peculiarities that make it different to the previously sketched intput graph.

The structure of edges remains the same, but obviously, paths are not forced to consist of permutations of the same tokens (as far as we are interested into multiple translation hypotheses), and there may also exist paths which do not reach the ending node n_N. These latter paths are not useful in rescoring tasks, but allowed in order to facilitate the study of the search graph. However, a very easy and efficient algorithm ($O(n)$, being n the search size) can be used in order to discard them, before rescoring work. Additionally, given that partial model costs are needed in rescoring work, our decoder allows to output the individual model costs computed for each translation unit (token t). Costs are encoded within the token s, as in the next example:

```
(0 (1 "o#or{1.5,0.9,0.6,0.2}" 6))
```

where the token t is now composed of the translation unit '*o#or*', followed by (four) model costs.

Multiple translation hypotheses can only be extracted if hypotheses recombinations are carefully saved. As in (Koehn, 2004), the decoder takes a record of any recombined hypothesis, allowing for a rigorous N-best generation. Model costs are referred to the current unit while the global score s is accumulated. Notice also that translation units (not words) are now used as tokens.

5 Experiments

Experiments are carried out for a Spanish-to-English translation task using the EPPS data set, corresponding to session transcriptions of the European Parliament.

Eff.	base	+tpos	+reor	+spos
Beam size $= 50$				
w/o cache	$1,820$	$2,170$	$2,970$	$3,260$
w/ cache	-50	-110	-190	-210
Beam size $= 100$				
w/o cache	$2,900$	$4,350$	$5,960$	$6,520$
w/ cache	-175	-410	-625	-640

Table 1: *Translation efficiency results.*

Table 1 shows translation efficiency results (measured in seconds) given two different beam search sizes. **w/cache** and **w/o cache** indicate whether the decoder employs (or not) the cache technique (section 3.3). Several system configuration have been tested: a baseline monotonous system using a 4-gram translation LM and a 5-gram target LM (**base**), extended with a target POS-tagged 5-gram LM (**+tpos**), further extended by allowing for reordering (**+reor**), and finally using a source-side POS-tagged 5-gram LM (**+spos**).

As it can be seen, the cache technique improves the efficiency of the search in terms of decoding time. Time results are further decreased (reduced time is shown for the **w/ cache** setting) by using more N-gram LM and allowing for a larger search graph (increasing the beam size and introducing distortion).

Further details on the previous experiment can be seen in (Crego and Mariño, 2006b; Crego and Mariño, 2006a), where additionally, the input word graph and extended N-gram tagged LM's are successfully used to improve accuracy at a very low computational cost.

Several publications can also be found in bibliography which show the use of output graphs in rescoring tasks allowing for clear accuracy improvements.

6 Conclusions

We have presented several extensions to MARIE, a freely available N-gram-based decoder. The extensions consist of accepting and generating word graphs, and introducing two N-gram LM's over source and target tagged words. Additionally, a caching technique is applied over the N-gram LM's.

Acknowledgments

This work has been funded by the European Union under the integrated project TC-STAR - (IST-2002-FP6-5067-38), the Spanish Government under the project AVIVAVOZ - (TEC2006-13694-C03) and the Universitat Politècnica de Catalunya under UPC-RECERCA grant.

References

J.M. Crego and J.B. Mariño. 2006a. Integration of postag-based source reordering into smt decoding by an extended search graph. *Proc. of the 7th Conf. of the Association for Machine Translation in the Americas*, pages 29–36, August.

J.M. Crego and J.B. Mariño. 2006b. Reordering experiments for n-gram-based smt. *1st IEEE/ACL Workshop on Spoken Language Technology*, December.

J.M. Crego, J.B. Mariño, and A. de Gispert. 2005. An ngram-based statistical machine translation decoder. *Proc. of the 9th European Conference on Speech Communication and Technology, Interspeech'05*, pages 3193–3196, September.

Ph. Koehn. 2004. Pharaoh: a beam search decoder for phrase-based statistical machine translation models. *Proc. of the 6th Conf. of the Association for Machine Translation in the Americas*, pages 115–124, October.

J.B. Mariño, R.E. Banchs, J.M. Crego, A. de Gispert, P. Lambert, J.A.R. Fonollosa, and M.R. Costa-jussà. 2006. N-gram based machine translation. *Computational Linguistics*, 32(4):527–549.

A Hybrid Approach to Word Segmentation and POS Tagging

Tetsuji Nakagawa
Oki Electric Industry Co., Ltd.
2−5−7 Honmachi, Chuo-ku
Osaka 541−0053, Japan
nakagawa378@oki.com

Kiyotaka Uchimoto
National Institute of Information and
Communications Technology
3−5 Hikaridai, Seika-cho, Soraku-gun
Kyoto 619−0289, Japan
uchimoto@nict.go.jp

Abstract

In this paper, we present a hybrid method for word segmentation and POS tagging. The target languages are those in which word boundaries are ambiguous, such as Chinese and Japanese. In the method, word-based and character-based processing is combined, and word segmentation and POS tagging are conducted simultaneously. Experimental results on multiple corpora show that the integrated method has high accuracy.

1 Introduction

Part-of-speech (POS) tagging is an important task in natural language processing, and is often necessary for other processing such as syntactic parsing. English POS tagging can be handled as a sequential labeling problem, and has been extensively studied. However, in Chinese and Japanese, words are not separated by spaces, and word boundaries must be identified before or during POS tagging. Therefore, POS tagging cannot be conducted without word segmentation, and how to combine these two processing is an important issue. A large problem in word segmentation and POS tagging is the existence of unknown words. Unknown words are defined as words that are not in the system's word dictionary. It is difficult to determine the word boundaries and the POS tags of unknown words, and unknown words often cause errors in these processing.

In this paper, we study a hybrid method for Chinese and Japanese word segmentation and POS tagging, in which word-based and character-based processing is combined, and word segmentation and POS tagging are conducted simultaneously. In the method, word-based processing is used to handle known words, and character-based processing is used to handle unknown words. Furthermore, information of word boundaries and POS tags are used at the same time with this method. The following sections describe the hybrid method and results of experiments on Chinese and Japanese corpora.

2 Hybrid Method for Word Segmentation and POS Tagging

Many methods have been studied for Chinese and Japanese word segmentation, which include word-based methods and character-based methods. Nakagawa (2004) studied a method which combines a word-based method and a character-based method. Given an input sentence in the method, a lattice is constructed first using a word dictionary, which consists of word-level nodes for all the known words in the sentence. These nodes have POS tags. Then, character-level nodes for all the characters in the sentence are added into the lattice (Figure 1). These nodes have position-of-character (POC) tags which indicate word-internal positions of the characters (Xue, 2003). There are four POC tags, B, I, E and S, each of which respectively indicates the beginning of a word, the middle of a word, the end of a word, and a single character word. In the method, the word-level nodes are used to identify known words, and the character-level nodes are used to identify unknown words, because generally word-level information is precise and appropriate for processing known words, and character-level information is robust and appropriate for processing unknown words. Extended hidden Markov models are used to choose the best path among all the possible candidates in the lattice, and the correct path is indicated by the thick lines in Figure 1. The POS tags and the POC tags are treated equally in the method. Thus, the word-level nodes and the character-level nodes are processed uniformly, and known words and unknown words are identified simultaneously. In the method, POS tags of known words as well as word boundaries are identified, but POS tags of unknown words are not identified. Therefore, we extend the method in order to conduct unknown word POS tagging too:

Hybrid Method

The method uses subdivided POC-tags in order to identify not only the positions of characters but also the parts-of-speech of the composing words (Figure 2, A). In the method, POS tagging of unknown words is conducted at the same time as word segmentation and POS tag-

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 217–220,
Prague, June 2007. ©2007 Association for Computational Linguistics

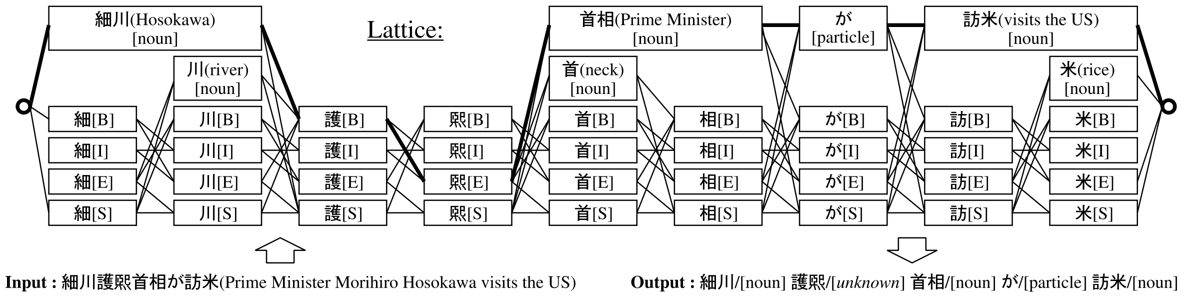


Figure 1: Word Segmentation and Known Word POS Tagging using Word and Character-based Processing

ging of known words, and information of parts-of-speech of unknown words can be used for word segmentation.

There are also two other methods capable of conducting unknown word POS tagging (Ng and Low, 2004):

Word-based Post-Processing Method
This method receives results of word segmentation and known word POS tagging, and predicts POS tags of unknown words using words as units (Figure 2, B). This approach is the same as the approach widely used in English POS tagging. In the method, the process of unknown word POS tagging is separated from word segmentation and known word POS tagging, and information of parts-of-speech of unknown words cannot be used for word segmentation. In later experiments, maximum entropy models were used deterministically to predict POS tags of unknown words. As features for predicting the POS tag of an unknown word w, we used the preceding and the succeeding two words of w and their POS tags, the prefixes and the suffixes of up to two characters of w, the character types contained in w, and the length of w.

Character-based Post-Processing Method
This method is similar to the word-based post-processing method, but in this method, POS tags of unknown words are predicted using characters as units (Figure 2, C). In the method, POS tags of unknown words are predicted using exactly the same probabilistic models as the hybrid method, but word boundaries and POS tags of known words are fixed in the post-processing step.

Ng and Low (2004) studied Chinese word segmentation and POS tagging. They compared several approaches, and showed that character-based approaches had higher accuracy than word-based approaches, and that conducting word segmentation and POS tagging all at once performed better than conducting these processing separately. Our hybrid method is similar to their character-based all-at-once approach. However, in their experiments, only word-based and character-based methods were examined. In our experiments, the combined method of word-based and character-based processing was examined. Furthermore, although their experiments were conducted with only Chinese data, we conducted experiments with Chinese and Japanese data, and confirmed that the hybrid method performed well on the Japanese data as well as the Chinese data.

3 Experiments

We used five word-segmented and POS-tagged corpora; the Penn Chinese Treebank corpus 2.0 (CTB), a part of the PFR corpus (PFR), the EDR corpus (EDR), the Kyoto University corpus version 2 (KUC) and the RWCP corpus (RWC). The first two were Chinese (C) corpora, and the rest were Japanese (J) corpora, and they were split into training and test data. The dictionary distributed with JUMAN version 3.61 (Kurohashi and Nagao, 1998) was used as a word dictionary in the experiments with the KUC corpus, and word dictionaries were constructed from all the words in the training data in the experiments with other corpora. Table 1 summarizes statistical information of the corpora: the language, the number of POS tags, the sizes of training and test data, and the splitting methods of them[1]. We used the following scoring measures to evaluate performance of word segmentation and POS tagging:

R : Recall (The ratio of the number of correctly segmented/POS-tagged words in system's output to the number of words in test data),

P : Precision (The ratio of the number of correctly segmented/POS-tagged words in system's output to the number of words in system's output),

[1] The unknown word rate for word segmentation is not equal to the unknown word rate for POS tagging in general, since the word forms of some words in the test data may exist in the word dictionary but the POS tags of them may not exist. Such words are regarded as known words in word segmentation, but as unknown words in POS tagging.

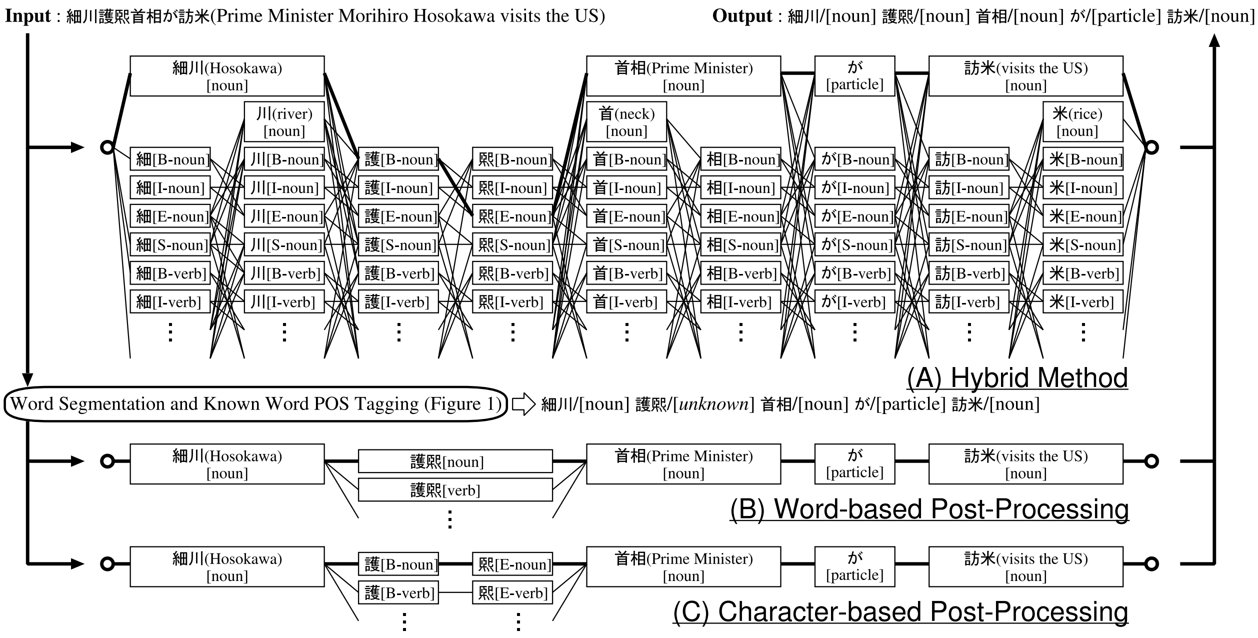

Input : 細川護熙首相が訪米(Prime Minister Morihiro Hosokawa visits the US)
Output : 細川/[noun] 護熙/[noun] 首相/[noun] が/[particle] 訪米/[noun]
細川(Hosokawa) [noun]
川(river) [noun]
首相(Prime Minister) [noun]
首(neck) [noun]
が [particle]
訪米(visits the US) [noun]
米(rice) [noun]
細[B-noun] 細[I-noun] 細[E-noun] 細[S-noun] 細[B-verb] 細[I-verb]
川[B-noun] 川[I-noun] 川[E-noun] 川[S-noun] 川[B-verb] 川[I-verb]
護[B-noun] 護[I-noun] 護[E-noun] 護[S-noun] 護[B-verb] 護[I-verb]
熙[B-noun] 熙[I-noun] 熙[E-noun] 熙[S-noun] 熙[B-verb] 熙[I-verb]
首[B-noun] 首[I-noun] 首[E-noun] 首[S-noun] 首[B-verb] 首[I-verb]
相[B-noun] 相[I-noun] 相[E-noun] 相[S-noun] 相[B-verb] 相[I-verb]
が[B-noun] が[I-noun] が[E-noun] が[S-noun] が[B-verb] が[I-verb]
訪[B-noun] 訪[I-noun] 訪[E-noun] 訪[S-noun] 訪[B-verb] 訪[I-verb]
米[B-noun] 米[I-noun] 米[E-noun] 米[S-noun] 米[B-verb] 米[I-verb]
(A) Hybrid Method
Word Segmentation and Known Word POS Tagging (Figure 1)
細川/[noun] 護熙/[unknown] 首相/[noun] が/[particle] 訪米/[noun]
護熙[noun]
護熙[verb]
(B) Word-based Post-Processing
護[B-noun] 熙[E-noun]
護[B-verb] 熙[E-verb]
(C) Character-based Post-Processing

Figure 2: Three Methods for Word Segmentation and POS Tagging

F : F-measure ($F = 2 \times R \times P/(R + P)$),
$R_{unknown}$: Recall for unknown words,
R_{known} : Recall for known words.

Table 2 shows the results[2]. In the table, **Word-based Post-Proc.**, **Char.-based Post-Proc.** and **Hybrid Method** respectively indicate results obtained with the word-based post-processing method, the character-based post-processing method, and the hybrid method. Two types of performance were measured: performance of word segmentation alone, and performance of both word segmentation and POS tagging. We first compare performance of both word segmentation and POS tagging. The F-measures of the hybrid method were highest on all the corpora. This result agrees with the observation by Ng and Low (2004) that higher accuracy was obtained by conducting word segmentation and POS tagging at the same time than by conducting these processing separately. Comparing the word-based and the character-based post-processing methods, the F-measures of the latter were higher on the Chinese corpora as reported by Ng and Low (2004), but the F-measures of the former were slightly higher on the Japanese corpora. The same tendency existed in the recalls for known words; the recalls of the character-based post-processing method were highest on the Chinese corpora, but those of the word-based method were highest on the Japanese corpora, except on the EDR corpus. Thus, the character-based method was not always better than the word-based method as reported by Ng and Low (2004) when the methods were used with the word and character-based combined approach on Japanese corpora. We next compare performance of word segmentation alone. The F-measures of the hybrid method were again highest in all the corpora, and the performance of word segmentation was improved by the integrated processing of word segmentation and POS tagging. The precisions of the hybrid method were highest with statistical significance on four of the five corpora. In all the corpora, the recalls for unknown words of the hybrid method were highest, but the recalls for known words were lowest.

Comparing our results with previous work is not easy since experimental settings are not the same. It was reported that the original combined method of word-based and character-based processing had high overall accuracy (F-measures) in Chinese word segmentation, compared with the state-of-the-art methods (Nakagawa, 2004). Kudo et al. (2004) studied Japanese word segmentation and POS tagging using conditional random fields (CRFs) and rule-based unknown word processing. They conducted experiments with the KUC corpus, and achieved F-measure of 0.9896 in word segmentation, which is better than ours (0.9847). Some features we did not used, such as base forms and conjugated forms of words, and hierarchical POS tags, were used in

[2] The recalls for known words of the word-based and the character-based post-processing methods differ, though the POS tags of known words are identified in the first common step. This is because known words are sometimes identified as unknown words in the first step and their POS tags are predicted in the post-processing step.

Corpus (Lang.)	Number of POS Tags	Number of Words (Unknown Word Rate for Segmentation/Tagging) [partition in the corpus]	
		Training	Test
CTB (C)	34	84,937 [sec. 1–270]	7,980 (0.0764 / 0.0939) [sec. 271–300]
PFR (C)	41	304,125 [Jan. 1–Jan. 9]	370,627 (0.0667 / 0.0749) [Jan. 10–Jan. 19]
EDR (J)	15	2,550,532 [$id = 4n + 0, id = 4n + 1$]	1,280,057 (0.0176 / 0.0189) [$id = 4n + 2$]
KUC (J)	40	198,514 [Jan. 1–Jan. 8]	31,302 (0.0440 / 0.0517) [Jan. 9]
RWC (J)	66	487,333 [1–10,000th sentences]	190,571 (0.0513 / 0.0587) [10,001–14,000th sentences]

Table 1: Statistical Information of Corpora

Corpus (Lang.)	Scoring Measure	Word Segmentation			Word Segmentation & POS Tagging		
		Word-based Post-Proc.	Char.-based Post-Proc.	Hybrid Method	Word-based Post-Proc.	Char.-based Post-Proc.	Hybrid Method
CTB (C)	R	0.9625	0.9625	**0.9639**	0.8922	0.8935	**0.8944**
	P	0.9408	0.9408	**0.9519***	0.8721	0.8733	**0.8832**
	F	0.9516	0.9516	**0.9578**	0.8821	0.8833	**0.8887**
	$R_{unknown}$	0.6492	0.6492	**0.7148**	0.4219	0.4312	**0.4713**
	R_{known}	**0.9885**	**0.9885**	0.9845	0.9409	**0.9414**	0.9382
PFR (C)	R	0.9503	0.9503	**0.9516**	0.8967	0.8997	**0.9024***
	P	0.9419	0.9419	**0.9485***	0.8888	0.8917	**0.8996***
	F	0.9461	0.9461	**0.9500**	0.8928	0.8957	**0.9010**
	$R_{unknown}$	0.6063	0.6063	**0.6674**	0.3845	0.3980	**0.4487**
	R_{known}	**0.9749**	**0.9749**	0.9719	0.9382	**0.9403**	0.9392
EDR (J)	R	**0.9525**	**0.9525**	**0.9525**	**0.9358**	0.9356	0.9357
	P	0.9505	0.9505	**0.9513***	0.9337	0.9335	**0.9346**
	F	0.9515	0.9515	**0.9519**	0.9347	0.9345	**0.9351**
	$R_{unknown}$	0.4454	0.4454	**0.4630**	0.4186	0.4103	**0.4296**
	R_{known}	**0.9616**	**0.9616**	0.9612	**0.9457**	**0.9457**	0.9454
KUC (J)	R	**0.9857**	**0.9857**	0.9850	0.9572	0.9567	**0.9574**
	P	0.9835	0.9835	**0.9843**	0.9551	0.9546	**0.9566**
	F	0.9846	0.9846	**0.9847**	0.9562	0.9557	**0.9570**
	$R_{unknown}$	0.9237	0.9237	**0.9302**	0.6724	0.6774	**0.6879**
	R_{known}	**0.9885**	**0.9885**	0.9876	**0.9727**	0.9719	0.9721
RWC (J)	R	0.9574	0.9574	**0.9592**	0.9225	0.9220	**0.9255***
	P	0.9533	0.9533	**0.9577***	0.9186	0.9181	**0.9241***
	F	0.9553	0.9553	**0.9585**	0.9205	0.9201	**0.9248**
	$R_{unknown}$	0.6650	0.6650	**0.7214**	0.4941	0.4875	**0.5467**
	R_{known}	**0.9732**	**0.9732**	0.9720	**0.9492**	0.9491	0.9491

(Statistical significance tests were performed for R and P, and * indicates significance at $p < 0.05$)

Table 2: Performance of Word Segmentation and POS Tagging

their study, and it may be a reason of the difference. Although, in our experiments, extended hidden Markov models were used to find the best solution, the performance will be further improved by using CRFs instead, which can easily incorporate a wide variety of features.

4 Conclusion

In this paper, we studied a hybrid method in which word-based and character-based processing is combined, and word segmentation and POS tagging are conducted simultaneously. We compared its performance of word segmentation and POS tagging with other methods in which POS tagging is conducted as a separated post-processing. Experimental results on multiple corpora showed that the hybrid method had high accuracy in Chinese and Japanese.

References

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Applying Conditional Random Fields to Japanese Morphological Analysis. In *Proceedings of EMNLP 2004*, pages 230–237.

Sadao Kurohashi and Makoto Nagao. 1998. *Japanese Morphological Analysis System JUMAN version 3.61*. Department of Informatics, Kyoto University. (in Japanese).

Tetsuji Nakagawa. 2004. Chinese and Japanese Word Segmentation Using Word-Level and Character-Level Information. In *Proceedings of COLING 2004*, pages 466–472.

Hwee Tou Ng and Jin Kiat Low. 2004. Chinese Part-of-Speech Tagging: One-at-a-Time or All-at-Once? Word-Based or Character-Based? In *Proceedings of EMNLP 2004*, pages 277–284.

Nianwen Xue. 2003. Chinese Word Segmentation as Character Tagging. *International Journal of Computational Linguistics and Chinese*, 8(1):29–48.

Automatic Part-of-Speech Tagging for Bengali: An Approach for Morphologically Rich Languages in a Poor Resource Scenario

Sandipan Dandapat, Sudeshna Sarkar, Anupam Basu
Department of Computer Science and Engineering
Indian Institute of Technology Kharagpur
India 721302
{sandipan,sudeshna,anupam.basu}@cse.iitkgp.ernet.in

Abstract

This paper describes our work on building Part-of-Speech (POS) tagger for Bengali. We have use Hidden Markov Model (HMM) and Maximum Entropy (ME) based stochastic taggers. Bengali is a morphologically rich language and our taggers make use of morphological and contextual information of the words. Since only a small labeled training set is available (45,000 words), simple stochastic approach does not yield very good results. In this work, we have studied the effect of using a morphological analyzer to improve the performance of the tagger. We find that the use of morphology helps improve the accuracy of the tagger especially when less amount of tagged corpora are available.

1 Introduction

Part-of-Speech (POS) taggers for natural language texts have been developed using linguistic rules, stochastic models as well as a combination of both (hybrid taggers). Stochastic models (Cutting et al., 1992; Dermatas et al., 1995; Brants, 2000) have been widely used in POS tagging for simplicity and language independence of the models. Among stochastic models, bi-gram and tri-gram Hidden Markov Model (HMM) are quite popular. Development of a high accuracy stochastic tagger requires a large amount of annotated text. Stochastic taggers with more than 95% word-level accuracy have been developed for English, German and other European Languages, for which large labeled data is available. Our aim here is to develop a stochastic POS tagger for Bengali but we are limited by lack of a large annotated corpus for Bengali. Simple HMM models do not achieve high accuracy when the training set is small. In such cases, additional information may be coded into the HMM model to achieve higher accuracy (Cutting et al., 1992). The semi-supervised model described in Cutting et al. (1992), makes use of both labeled training text and some amount of unlabeled text. Incorporating a diverse set of overlapping features in a HMM-based tagger is difficult and complicates the smoothing typically used for such taggers. In contrast, methods based on Maximum Entropy (Ratnaparkhi, 1996), Conditional Random Field (Shrivastav, 2006) etc. can deal with diverse, overlapping features.

1.1 Previous Work on Indian Language POS Tagging

Although some work has been done on POS tagging of different Indian languages, the systems are still in their infancy due to resource poverty. Very little work has been done previously on POS tagging of Bengali. Bengali is the main language spoken in Bangladesh, the second most commonly spoken language in India, and the fourth most commonly spoken language in the world. Ray et al. (2003) describes a morphology-based disambiguation for Hindi POS tagging. System using a decision tree based learning algorithm (CN2) has been developed for statistical Hindi POS tagging (Singh et al., 2006). A reasonably good accuracy POS tagger for Hindi has been developed using Maximum Entropy Markov Model (Dalal et al., 2007). The system uses linguistic suffix and POS categories of a word along with other contextual features.

2 Our Approach

The problem of POS tagging can be formally stated as follows. Given a sequence of words w_1 … w_n, we want to find the corresponding sequence of tags $t_1 \ldots t_n$, drawn from a set of tags T. We use a tagset of 40 tags[1]. In this work, we explore supervised and semi-supervised bi-gram

[1] http://www.mla.iitkgp.ernet.in/Tag.html

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 221–224,
Prague, June 2007.

HMM and a ME based model. The bi-gram assumption states that the POS-tag of a word depends on the current word and the POS tag of the previous word. An ME model estimates the probabilities based on the imposed constraints. Such constraints are derived from the training data, maintaining some relationship between features and outcomes. The most probable tag sequence for a given word sequence satisfies equation (1) and (2) respectively for HMM and ME model:

$$S = \arg\max_{t_1 \ldots t_n} \prod_{i=1,n} P(w_i \mid t_i) P(t_i \mid t_{i-1}) \qquad (1)$$

$$p(t_1 \ldots t_n \mid w_1 \ldots w_n) = \prod_{i=1,n} p(t_i \mid h_i) \qquad (2)$$

Here, h_i is the context for word w_i. Since the basic bigram model of HMM as well as the equivalent ME models do not yield satisfactory accuracy, we wish to explore whether other available resources like a morphological analyzer can be used appropriately for better accuracy.

2.1 HMM and ME based Taggers

Three taggers have been implemented based on bigram HMM and ME model. The first tagger (we shall call it **HMM-S**) makes use of the supervised HMM model parameters, whereas the second tagger (we shall call it **HMM-SS**) uses the semi supervised model parameters. The third tagger uses **ME** based model to find the most probable tag sequence for a given sequence of words.

In order to further improve the tagging accuracy, we use a Morphological Analyzer (MA) and integrate morphological information with the models. We assume that the POS-tag of a word *w* can take values from the set $T_{MA}(w)$, where $T_{MA}(w)$ is computed by the Morphological Analyzer. Note that the size of $T_{MA}(w)$ is much smaller than T. Thus, we have a restricted choice of tags as well as tag sequences for a given sentence. Since the correct tag *t* for *w* is always in $T_{MA}(w)$ (assuming that the morphological analyzer is complete), it is always possible to find out the correct tag sequence for a sentence even after applying the morphological restriction. Due to a much reduced set of possibilities, this model is expected to perform better for both the HMM (HMM-S and HMM-SS) and ME models even when only a small amount of labeled training text is available. We shall call these new models **HMM-S+MA**, **HMM-SS+ MA** and **ME+MA**.

Our MA has high accuracy and coverage but it still has some missing words and a few errors. For the purpose of these experiments we have made sure that all words of the test set are present in the root dictionary that an MA uses.

While MA helps us to restrict the possible choice of tags for a given word, one can also use suffix information (i.e., the sequence of last few characters of a word) to further improve the models. For HMM models, suffix information has been used during smoothing of emission probabilities, whereas for ME models, suffix information is used as another type of feature. We shall denote the models with suffix information with a **'+suf'** marker. Thus, we have – **HMM-S+suf**, **HMM-S+suf+MA**, **HMM-SS+suf** etc.

2.1.1 Unknown Word Hypothesis in HMM

The transition probabilities are estimated by linear interpolation of unigrams and bigrams. For the estimation of emission probabilities add-one smoothing or suffix information is used for the unknown words. If the word is unknown to the morphological analyzer, we assume that the POS-tag of that word belongs to any of the open class grammatical categories (all classes of Noun, Verb, Adjective, Adverb and Interjection).

2.1.2 Features of the ME Model

Experiments were carried out to find out the most suitable binary valued features for the POS tagging in the ME model. The main features for the POS tagging task have been identified based on the different possible combination of the available word and tag context. The features also include prefix and suffix up to length four. We considered different combinations from the following set for obtaining the best feature set for the POS tagging task with the data we have.

$$F = \{ w_i, w_{i-1}, w_{i-2}, w_{i+1}, w_{i+2}, t_{i-1}, t_{i-2}, |pre| \le 4, |suf| \le 4 \}$$

Forty different experiments were conducted taking several combinations from set '*F*' to identify the best suited feature set for the POS tagging task. From our empirical analysis we found that the combination of contextual features (current word and previous tag), prefixes and suffixes of length ≤ 4 gives the best performance for the ME model. It is interesting to note that the inclusion of prefix and suffix for all words gives better result instead of using only for rare words as is described in Ratnaparkhi (1996). This can be explained by the fact that due to small amount of annotated data, a significant number of instances are not found for most of the word of the language vocabulary.

3 Experiments

We have a total of 12 models as described in subsection 2.1 under different stochastic tagging schemes. The same training text has been used to estimate the parameters for all the models. The model parameters for supervised HMM and ME models are estimated from the annotated text corpus. For semi-supervised learning, the HMM learned through supervised training is considered as the initial model. Further, a larger unlabelled training data has been used to re-estimate the model parameters of the semi-supervised HMM. The experiments were conducted with three different sizes (10K, 20K and 40K words) of the training data to understand the relative performance of the models as we keep on increasing the size of the annotated data.

3.1 Training Data

The training data includes manually annotated 3625 sentences (approximately 40,000 words) for both supervised HMM and ME model. A fixed set of 11,000 unlabeled sentences (approximately 100,000 words) taken from CIIL corpus[2] are used to re-estimate the model parameter during semi-supervised learning. It has been observed that the corpus ambiguity (mean number of possible tags for each word) in the training text is 1.77 which is much larger compared to the European languages (Dermatas et al., 1995).

3.2 Test Data

All the models have been tested on a set of randomly drawn 400 sentences (5000 words) disjoint from the training corpus. It has been noted that 14% words in the open testing text are unknown with respect to the training set, which is also a little higher compared to the European languages (Dermatas et al., 1995)

3.3 Results

We define the tagging accuracy as the ratio of the correctly tagged words to the total number of words. Table 1 summarizes the final accuracies achieved by different learning methods with the varying size of the training data. Note that the baseline model (i.e., the tag probabilities depends only on the current word) has an accuracy of 76.8%.

Method	**Accuracy**		
	10K	**20K**	**40K**
HMM-S	57.53	70.61	77.29
HMM-S+suf	75.12	79.76	83.85
HMM-S+MA	82.39	84.06	86.64
HMM-S+suf+MA	84.73	87.35	88.75
HMM-SS	63.40	70.67	77.16
HMM-SS+suf	75.08	79.31	83.76
HMM-SS+MA	83.04	84.47	86.41
HMM-SS+suf+MA	84.41	87.16	87.95
ME	74.37	79.50	84.56
ME+suf	77.38	82.63	86.78
ME+MA	82.34	84.97	87.38
ME+suf+MA	84.13	87.07	88.41

Table 1: Tagging accuracies (in %) of different models with 10K, 20K and 40K training data.

3.4 Observations

We find that in both the HMM based models (**HMM-S** and **HMM-SS**), the use of suffix information as well as the use of a morphological analyzer improves the accuracy of POS tagging with respect to the base models. The use of MA gives better results than the use of suffix information. When we use both suffix information as well as MA, the results is even better.

HMM-SS does better than **HMM-S** when very little tagged data is available, for example, when we use 10K training corpus. However, the accuracy of the semi-supervised HMM models are slightly poorer than that of the supervised HMM models for moderate size training data and use of suffix information. This discrepancy arises due to the over-fitting of the supervised models in the case of small training data; the problem is alleviated with the increase in the annotated data.

As we have noted already the use of MA and/or suffix information improves the accuracy of the POS tagger. But what is significant to note is that the percentage of improvement is higher when the amount of training data is less. The **HMM-S+suf** model gives an improvement of around 18%, 9% and 6% over the **HMM-S** model for 10K, 20K and 40K training data respectively. Similar trends are observed in the case of the semi-supervised HMM and the ME models. The use of morphological restriction (**HMM-S+MA**) gives an improvement of 25%, 14% and 9% respectively over the **HMM-S** in case of 10K, 20K

[2] A part of the EMILE/CIIL corpus developed at Central Institute of Indian Languages (CIIL), Mysore.

and 40K training data. As the improvement due to MA decreases with increasing data, it might be concluded that the use of morphological restriction may not improve the accuracy when a large amount of training data is available. From our empirical observations we found that both suffix and morphological restriction (**HMM-S+suf+MA**) gives an improvement of 27%, 17% and 12% over the HMM-S model respectively for the three different sizes of training data.

The Maximum Entropy model does better than the HMM models for smaller training data. But with higher amount of training data the performance of the HMM and ME model are comparable. Here also we observe that suffix information and MA have positive effect, and the effect is higher with poor resources.

Furthermore, in order to estimate the relative performance of the models, experiments were carried out with two existing taggers: TnT (Brants, 2000) and ACOPOST[3]. The accuracy achieved using TnT are 87.44% and 87.36% respectively with bigram and trigram model for 40K training data. The accuracy with ACOPOST is 86.3%. This reflects that the higher order Markov models do not work well under the current experimental setup.

3.5 Assessment of Error Types

Table 2 shows the top five confusion classes for HMM-S+MA model. The most common types of errors are the confusion between proper noun and common noun and the confusion between adjective and common noun. This results from the fact that most of the proper nouns can be used as common nouns and most of the adjectives can be used as common nouns in Bengali.

Actual Class (frequency)	Predicted Class	% of total errors	% of class errors
NP(251)	NN	21.03	43.82
JJ(311)	NN	5.16	8.68
NN(1483)	JJ	4.78	1.68
DTA(100)	PP	2.87	15.0
NN(1483)	VN	2.29	0.81

Table 2: Five most common types of errors

Almost all the confusions are wrong assignment due to less number of instances in the training corpora, including errors due to long distance phenomena.

4 Conclusion

In this paper we have described an approach for automatic stochastic tagging of natural language text for Bengali. The models described here are very simple and efficient for automatic tagging even when the amount of available annotated text is small. The models have a much higher accuracy than the naïve baseline model. However, the performance of the current system is not as good as that of the contemporary POS-taggers available for English and other European languages. The best performance is achieved for the supervised learning model along with suffix information and morphological restriction on the possible grammatical categories of a word. In fact, the use of MA in any of the models discussed above enhances the performance of the POS tagger significantly. We conclude that the use of morphological features is especially helpful to develop a reasonable POS tagger when tagged resources are limited.

References

A. Dalal, K. Nagaraj, U. Swant, S. Shelke and P. Bhattacharyya. 2007. *Building Feature Rich POS Tagger for Morphologically Rich Languages: Experience in Hindi*. ICON, 2007.

A. Ratnaparkhi, 1996. *A maximum entropy part-of-speech tagger*. EMNLP 1996. pp. 133-142.

D. Cutting, J. Kupiec, J. Pederson and P. Sibun. 1992. *A practical part-of-speech tagger*. In Proc. of the 3rd Conference on Applied NLP, pp. 133-140.

E. Dermatas and K. George. 1995. *Automatic stochastic tagging of natural language texts*. Computational Linguistics, 21(2): 137-163.

M. Shrivastav, R. Melz, S. Singh, K. Gupta and P. Bhattacharyya, 2006. *Conditional Random Field Based POS Tagger for Hindi*. In Proceedings of the MSPIL, pp. 63-68.

P. R. Ray, V. Harish, A. Basu and S. Sarkar, 2003. *Part of Speech Tagging and Local Word Grouping Techniques for Natural Language Processing*. ICON 2003.

S. Singh, K. Gupta, M. Shrivastav and P. Bhattacharyya, 2006. *Morphological Richness Offset Resource Demand – Experience in constructing a POS Tagger for Hindi*. COLING/ACL 2006, pp. 779-786.

T. Brants. 2000. TnT – *A statistical part-of-sppech tagger*. In Proc. of the 6th Applied NLP Conference, pp. 224-231.

[3] http://maxent.sourceforge.net

Japanese Dependency Parsing Using Sequential Labeling for Semi-spoken Language

Kenji Imamura and **Genichiro Kikui**
NTT Cyber Space Laboratories, NTT Corporation
1-1 Hikarinooka, Yokosuka-shi, Kanagawa, 239-0847, Japan
{imamura.kenji, kikui.genichiro}@lab.ntt.co.jp

Norihito Yasuda
NTT Communication Science Laboratories, NTT Corporation
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237, Japan
n-yasuda@cslab.kecl.ntt.co.jp

Abstract

The amount of documents directly published by end users is increasing along with the growth of Web 2.0. Such documents often contain spoken-style expressions, which are difficult to analyze using conventional parsers. This paper presents dependency parsing whose goal is to analyze Japanese semi-spoken expressions. One characteristic of our method is that it can parse self-dependent (independent) segments using sequential labeling.

1 Introduction

Dependency parsing is a way of structurally analyzing a sentence from the viewpoint of modification. In Japanese, relationships of modification between phrasal units called *bunsetsu* segments are analyzed. A number of studies have focused on parsing of Japanese as well as of other languages. Popular parsers are CaboCha (Kudo and Matsumoto, 2002) and KNP (Kurohashi and Nagao, 1994), which were developed to analyze formal written language expressions such as that in newspaper articles.

Generally, the syntactic structure of a sentence is represented as a tree, and parsing is carried out by maximizing the likelihood of the tree (Charniak, 2000; Uchimoto et al., 1999). Units that do not modify any other units, such as fillers, are difficult to place in the tree structure. Conventional parsers have forced such independent units to modify other units.

Documents published by end users (e.g., blogs) are increasing on the Internet along with the growth of Web 2.0. Such documents do not use controlled written language and contain fillers and emoticons. This implies that analyzing such documents is difficult for conventional parsers.

This paper presents a new method of Japanese dependency parsing that utilizes sequential labeling based on conditional random fields (CRFs) in order to analyze semi-spoken language. Concretely, sequential labeling assigns each segment a dependency label that indicates its relative position of dependency. If the label set includes self-dependency, the fillers and emoticons would be analyzed as segments depending on themselves. Therefore, since it is not necessary for the parsing result to be a tree, our method is suitable for semi-spoken language.

2 Methods

Japanese dependency parsing for written language is based on the following principles. Our method relaxes the first principle to allow self-dependent segments (c.f. Section 2.3).

1. Dependency moves from left to right.
2. Dependencies do not cross each other.
3. Each segment, except for the top of the parsed tree, modifies at most one other segment.

2.1 Dependency Parsing Using Cascaded Chunking (CaboCha)

Our method is based on the cascaded chunking method (Kudo and Matsumoto, 2002) proposed as the CaboCha parser [1]. CaboCha is a sort of shift-reduce parser and determines whether or not a segment depends on the next segment by using an

[1] http://www.chasen.org/˜taku/software/cabocha/

Proceedings of the ACL 2007 Demo and Poster Sessions, pages 225–228,
Prague, June 2007. ©2007 Association for Computational Linguistics

SVM-based classifier. To analyze long-distance dependencies, CaboCha shortens the sentence by removing segments for which dependencies are already determined and which no other segments depend on. CaboCha constructs a tree structure by repeating the above process.

2.2 Sequential Labeling

Sequential labeling is a process that assigns each unit of an input sequence an appropriate label (or tag). In natural language processing, it is applied to, for example, English part-of-speech tagging and named entity recognition. Hidden Markov models or conditional random fields (Lafferty et al., 2001) are used for labeling. In this paper, we use linear-chain CRFs.

In sequential labeling, training data developers can design labels with no restrictions.

2.3 Cascaded Chunking Using Sequential Labeling

The method proposed in this paper is a generalization of CaboCha. Our method considers not only the next segment, but also the following N segments to determine dependencies. This area, including the considered segment, is called the *window*, and N is called the window size. The parser assigns each segment a dependency label that indicates where the segment depends on the segments in the window. The flow is summarized as follows:

1. Extract features from segments such as the part-of-speech of the headword in a segment (c.f. Section 3.1).
2. Carry out sequential labeling using the above features.
3. Determine the actual dependency by interpreting the labels.
4. Shorten the sentence by deleting segments for which the dependency is already determined and that other segments have never depended on.
5. If only one segment remains, then finish the process. If not, return to Step 1.

An example of dependency parsing for written language is shown in Figure 1 (a).

In Steps 1 and 2, dependency labels are supplied to each segment in a way similar to that used by other sequential labeling methods. However, our sequential labeling has the following characteristics since this task is dependency parsing.

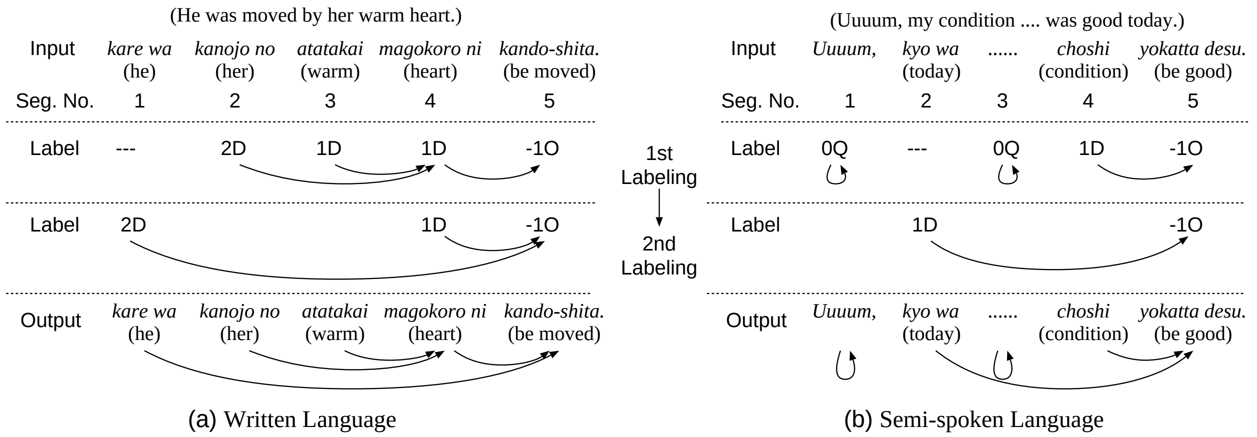
Label	Description
—	Segment depends on a segment outside of window.
0Q	Self-dependency
1D	Segment depends on next segment.
2D	Segment depends on segment after next.
-1O	Segment is top of parsed tree.

Table 1: Label List Used by Sequential Labeling (Window Size: 2)

- The labels indicate relative positions of the dependent segment from the current segment (Table 1). Therefore, the number of labels changes according to the window size. Long-distance dependencies can be parsed by one labeling process if we set a large window size. However, growth of label variety causes data sparseness problems.
- One possible label is that of self-dependency (noted as '0Q' in this paper). This is assigned to independent segments in a tree.
- Also possible are two special labels. Label '-1O' denotes a segment that is the top of the parsed tree. Label '—' denotes a segment that depends on a segment outside of the window. When the window size is two, the segment depends on a segment that is over two segments ahead.
- The label for the current segment is determined based on all features in the window and on the label of the previous segment.

In Step 4, segments, which no other segments depend on, are removed in a way similar to that used by CaboCha. The principle that dependencies do not cross each other is applied in this step. For example, if a segment depends on a segment after the next, the next segment cannot be modified by other segments. Therefore, it can be removed. Similarly, since the '—' label indicates that the segment depends on a segment after N segments, all intermediate segments can be removed if they do not have '—' labels.

The sentence is shortened by iteration of the above steps. The parsing finishes when only one segment remains in the sentence (this is the segment



(a) Written Language

(b) Semi-spoken Language

Figure 1: Examples of Dependency Parsing (Window Size: 2)

Corpus	Type	# of Sentences	# of Segments
Kyoto	Training	24,283	234,685
	Test	9,284	89,874
Blog	Training	18,163	106,177
	Test	8,950	53,228

Table 2: Corpus Size

at the top of the parsed tree). In the example in Figure 1 (a), the process finishes in two iterations.

In a sentence containing fillers, the self-dependency labels are assigned by sequential labeling, as shown in Figure 1 (b), and are parsed as independent segments. Therefore, our method is suitable for parsing semi-spoken language that contains independent segments.

3 Experiments

3.1 Experimental Settings

Corpora In our experiments, we used two corpora. One is the Kyoto Text Corpus 4.0 [2], which is a collection of newspaper articles with segment and dependency annotations. The other is a blog corpus, which is a collection of blog articles taken as semi-spoken language. The blog corpus is manually annotated in a way similar to that used for the Kyoto text corpus. The sizes of the corpora are shown in Table 2.

Training We used CRF++ [3], a linear-chain CRF training tool, with eleven features per segment. All of these are static features (proper to each segment) such as surface forms, parts-of-speech, inflections of a content headword and a functional headword in a segment. These are parts of a feature set that many papers have referenced (Uchimoto et al., 1999; Kudo and Matsumoto, 2002).

Evaluation Metrics Dependency accuracy and sentence accuracy were used as evaluation metrics. Sentence accuracy is the proportion of total sentences in which all dependencies in the sentence are accurately labeled. In Japanese, the last segment of most sentences is the top of the parsed trees, and many papers exclude this last segment from the accuracy calculation. We, in contrast, include the last one because some of the last segments are self-dependent.

3.2 Accuracy of Dependency Parsing

Dependency parsing was carried out by combining training and test corpora. We used a window size of three. We also used CaboCha as a reference for the set of sentences trained only with the Kyoto corpus because it is designed for written language. The results are shown in Table 3.

CaboCha had better accuracies for the Kyoto test corpus. One reason might be that our method manually combined features and used parts of combinations, while CaboCha automatically finds the best combinations by using second-order polynomial kernels.

For the blog test corpus, the proposed method using the Kyoto+Blog model had the best depen-

[2] http://nlp.kuee.kyoto-u.ac.jp/nl-resource/corpus.html

[3] http://www.chasen.org/˜taku/software/CRF++/

Test Corpus	Method	Training Corpus (Model)	Dependency Accuracy	Sentence Accuracy
Kyoto (Written Language)	Proposed Method (Window Size: 3)	Kyoto	89.87% (80766 / 89874)	48.12% (4467 / 9284)
		Kyoto + Blog	89.76% (80670 / 89874)	47.63% (4422 / 9284)
	CaboCha	Kyoto	**92.03**% (82714 / 89874)	**55.36**% (5140 / 9284)
Blog (Semi-spoken Language)	Proposed Method (Window Size: 3)	Kyoto	77.19% (41083 / 53226)	41.41% (3706 / 8950)
		Kyoto + Blog	**84.59**% (45022 / 53226)	**52.72**% (4718 / 8950)
	CaboCha	Kyoto	77.44% (41220 / 53226)	43.45% (3889 / 8950)

Table 3: Dependency and Sentence Accuracies among Methods/Corpora

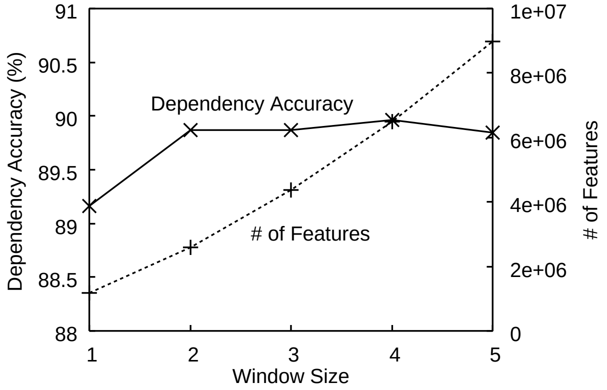


Figure 2: Dependency Accuracy and Number of Features According to Window Size (The Kyoto Text Corpus was used for training and testing.)

dency accuracy result at 84.59%. This result was influenced not only by the training corpus that contains the blog corpus but also by the effect of self-dependent segments. The blog test corpus contains 3,089 self-dependent segments, and 2,326 of them (75.30%) were accurately parsed. This represents a dependency accuracy improvement of over 60% compared with the Kyoto model.

Our method is effective in parsing blogs because fillers and emoticons can be parsed as self-dependent segments.

3.3 Accuracy According to Window Size

Another characteristic of our method is that all dependencies, including long-distance ones, can be parsed by one labeling process if the window covers the entire sentence. To analyze this characteristic, we evaluated dependency accuracies in various window sizes. The results are shown in Figure 2.

The number of features used for labeling increases exponentially as window size increases. However, dependency accuracy was saturated after a window size of two, and the best accuracy was when the window size was four. This phenomenon implies a data sparseness problem.

4 Conclusion

We presented a new dependency parsing method using sequential labeling for the semi-spoken language that frequently appears in Web documents. Sequential labeling can supply segments with flexible labels, so our method can parse independent words as self-dependent segments. This characteristic affects robust parsing when sentences contain fillers and emoticons.

The other characteristics of our method are using CRFs and that long dependencies are parsed in one labeling process. SVM-based parsers that have the same characteristics can be constructed if we introduce multi-class classifiers. Further comparisons with SVM-based parsers are future work.

References

Eugene Charniak. 2000. A maximum-entropy-inspired parser. In *Proc. of NAACL-2000*, pages 132–139.

Taku Kudo and Yuji Matsumoto. 2002. Japanese dependency analyisis using cascaded chunking. In *Proc. of CoNLL-2002*, Taipei.

Sadao Kurohashi and Makoto Nagao. 1994. A syntactic analysis method of long Japanese sentences based on the detection of conjunctive structures. *Computational Linguistics*, 20(4):507–534.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML-2001*, pages 282–289.

Kiyotaka Uchimoto, Satoshi Sekine, and Hitoshi Isahara. 1999. Japanese dependency structure analysis based on maximum entropy models. In *Proc. of EACL'99*, pages 196–203, Bergen, Norway.

Author Index