HLT/EMNLP 2005

Interactive Demonstrations Proceedings

Co-chairs:

Donna Byron, The Ohio State University
Anand Venkataraman, SRI International
Dell Zhang, Birkbeck, University of London

October 7, 2005

Vancouver, British Columbia, Canada

Production and Manufactured by
Omnipress Inc.
Post Office Box 7214
Madison, WI 53707-7214

The conference organizers are grateful to the following sponsors for their generous support.

Silver Sponsor:

Google™

Bronze Sponsors:

elDa

Microsoft®

SHARP

XEROX®
Research Centre Europe

Sponsor of Best Student Paper Award:

IBM®

This year's HLT/EMNLP conference is co-sponsored by ***ACL SIGDAT****.*

Order copies of this and other ACL proceedings from:

Association for Computational Linguistics (ACL)
3 Landmark Center
East Stroudsburg, PA 18301
USA
Tel: +1-570-476-8006
Fax: +1-570-476-0860
acl@aclweb.org

Preface to the Demonstration Proceedings

This volume contains the abstracts of the technology demonstrations that were presented at the combined 2005 Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, held in Vancouver, British Columbia, Canada on October 7, 2005. The demonstration program has traditionally been an important component of the HLT conference, and the addition of EMNLP this year creates a new opportunity for interaction among researchers coming from a variety of methodological perspectives. The demonstration program allows researchers the opportunity to showcase new and innovative technology, to update the research community on ongoing projects, and to show software tools developed for corpus development or other research aids to the users they were designed to assist. Demo technologies included in the program were selected based on their potential appeal to the HLT/EMNLP audience, technical and innovative merit, and completeness as a stand-alone demo. Out of 31 demonstration proposals submitted for review, 20 were selected for inclusion at the conference, representing 32 institutions from Asia, North America, and Europe.

We would like to acknowledge Google, Inc. for their generous support of the conference and the demo session. We are also grateful to our reviewers for spending time to help us select a set of demonstrations that together compose a high-quality and informative demo program. We would also like to thank the HLT/EMNLP-2005 conference organizers for their assistance in setting up the program, and Priscilla Rasmussen for local organization.

Donna Byron, Anand Venkataraman and Dell Zhang (editors and co-chairs)
August, 2005

Review Committee:

Harry Bratt, SRI International
Hang Cui, National University of Singapore
Chris Culy, N/A
Marcello Federico, ITC-irst
Eric Fosler-Lussier, The Ohio State University
Andreas Kathol, SRI International
Xiaoli Li, Institute for Infocomm Research, Singapore
Bing Liu, University of Illinois at Chicago
Yang Liu, ICSI, Berkeley
Wen-Hsiang Lu, National Cheng Kung University, Taiwan
Detmar Meurers, The Ohio State University
Murat Saraclar, Bogazici University
Elizabeth Shriberg, SRI International
Gokhan Tur, AT&T Labs
Wen Wang, SRI International
Timothy Weale, The Ohio State University

Table of Contents

Conference Program

Friday, October 7, 2005

Session 1: 6:30-8:00pm

Automatic Detection of Translation Errors: The State of the Art
Graham Russell, George Foster and Ngoc Tran Nguyen

Bridging the Gap between Technology and Users: Leveraging Machine
Thomas Hoeft, Nick Cramer, M. L. Gregory and Elizabeth Hetzler

Classummary: Introducing Discussion Summarization to Online Classrooms
Liang Zhou, Erin Shaw, Chin-Yew Lin and Eduard Hovy

Demonstrating an Interactive Semantic Role Labeling System
Vasin Punyakanok, Dan Roth, Mark Sammons and Wen-tau Yih

MindNet: An Automatically-Created Lexical Resource
Lucy Vanderwende, Gary Kacmarcik, Hisami Suzuki and Arul Menezes

NooJ: a Linguistic Annotation System for Corpus Processing
Max Silberztein

Pattern Visualization for Machine Translation Output
Adam Lopez and Philip Resnik

Prague Dependency Treebank as an Exercise Book of Czech
Barbora Hladká and Ondřej Kučera

Translation Exercise Assistant: Automated Generation of Translation
Jill Burstein and Daniel Marcu

WebExperimenter for Multiple-Choice Question Generation
Ayako Hoshino and Hiroshi Nakagawa

Friday, October 7, 2005 (continued)

Session 2: 8:00-9:30pm

DialogueView: an Annotation Tool for Dialogue
Fan Yang and Peter A. Heeman

Extracting Information about Outbreaks of Infectious Epidemics
Roman Yangarber, Lauri Jokipii, Antti Rauramo and Silja Huttunen

A Flexible Conversational Dialog System for MP3 Player
Fuliang Weng, Lawrence Cavedon, Badri Raghunathan, Danilo Mirkovic, Ben Bei, Heather Pon-Barry, Harry Bratt, Hua Cheng, Hauke Schmidt, Rohit Mishra, Brian Lathrop, Qi Zhang, Tobias Scheideck, Kui Xu, Tess Hand-Bender, Stanley Peters, Liz Shriberg and Carsten Bergmann

Japanese Speech Understanding using Grammar Specialization
Manny Rayner, Nikos Chatzichrisafis, Pierrette Bouillon, Yukie Nakao, Hitoshi Isahara, Kyoko Kanzaki, Beth Ann Hockey, Marianne Santaholma and Marianne Starlander

The MIT Spoken Lecture Processing Project
James R. Glass, Timothy J. Hazen, D. Scott Cyphers, Ken Schutte and Alex Park

MBOI: Discovery of Business Opportunities on the Internet
Arman Tajarobi, Jean-François Garneau and François Paradis

OPINE: Extracting Product Features and Opinions from Reviews
Ana-Maria Popescu, Bao Nguyen and Oren Etzioni

OpinionFinder: A System for Subjectivity Analysis
Theresa Wilson, Paul Hoffmann, Swapna Somasundaran, Jason Kessler, Janyce Wiebe, Yejin Choi, Claire Cardie, Ellen Riloff and Siddharth Patwardhan

POSBIOTM/W: A Development Workbench for Machine Learning Oriented Biomedical Text Mining System
Kyungduk Kim, Yu Song and Gary Geunbae Lee

Automatic Detection of Translation Errors: The State of the Art

Graham Russell and **Ngoc Tran Nguyen**
IIT–ILT, National Research Council Canada
RALI-DIRO, Université de Montréal[*]
{russell,nguyentt}@iro.umontreal.ca

George Foster
IIT–ILT, National Research Council Canada[†]
george.foster@nrc-cnrc.gc.ca

1 Background

The demonstration presents TransCheck, a translation quality-assurance tool developed jointly by the RALI group at the University of Montreal and the Interactive Language Technologies section of the Canadian National Research Council's Institute for Information Technology.

The system differs from other similar tools in the range of error-types targeted, and the underlying mechanisms employed. The demonstration illustrates the operation of the system and gives the rationale for its design and capabilities. The version demonstrated accepts input in English and French.

2 System Overview

A modular architecture promotes flexibility (ease of adaptation to new domains, client requirements and language pairs) and extensibility (incorporation of new error-detector components as they become available).

In a transparent preprocessing stage, source and target texts are read and aligned. The resulting stream of alignment regions is passed to a set of independent error-detection modules, each of which records errors in a global table for subsequent report generation. Certain of the error-detection components make use of external data in the form of lexical and other language resources.

3 Translation Errors

The difficulty of general-case translation error detection is discussed. Several classes of feasible errors are identified, and the technological capabilities required for their successful detection described.

Detection of incorrect terminology usage, for example, requires the ability to recognize correspondences between source and target language expressions, and to generalize over different realizations of a given term; inflection, coordination and anaphora combine to render inadequate solutions based solely on simple static lists of term pairs. 'Negative terminology', covering false friends, deceptive cognates, Anglicisms, etc., is rather more challenging, and can benefit from a more precise notion of translational correspondence. Proper names pose a range of problems, including referential disambiguation and varying conventions regarding transliteration, while a broad class of paralinguistic phenomena (numbers, dates, product codes, etc.) raise yet others in the area of monolingual analysis and translational equivalence. Omissions and insertions constitute a final error class; these present particular difficulties of recognition and interpretation, and are best addressed heuristically.

The current TransCheck system targets the error types mentioned above. Each is exemplified and discussed, together with the elements of language technology which permit their detection: dictionaries, shallow parsing, alignment, translation models, etc.

Experience gained in preliminary user trials is briefly reported and a variety of usage scenarios considered. Finally, some comparisons are made with other translation tools, including other proposals for translation error detection.

[*] C.P. 6128, succ. Centre-ville, Montréal QC, Canada H3C 3J7

[†] University of Quebec en Outaouais, Lucien Brault Pavilion, 101 St-Jean-Bosco Street, Gatineau QC, Canada K1A 0R6

Proceedings of HLT/EMNLP 2005 Demonstration Abstracts, page 1,
Vancouver, October 2005.

Bridging the Gap between Technology and Users: Leveraging Machine Translation in a Visual Data Triage Tool

Thomas Hoeft
Pacific Northwest
National Laboratory
902 Battelle Blvd.
Richland, WA 99354

Nick Cramer
Pacific Northwest
National Laboratory
902 Battelle Blvd.
Richland, WA 99354

M. L. Gregory
Pacific Northwest
National Laboratory
902 Battelle Blvd.
Richland, WA 99354

Elizabeth Hetzler
Pacific Northwest
National Laboratory
902 Battelle Blvd.
Richland, WA 99354

{thomas.hoeft;nick.cramer;michelle.gregory;beth.hetzler}@pnl.gov

1 Introduction

While one of the oldest pursuits in computational linguistics (see Bar-Hillel, 1951), machine translation (MT) remains an unsolved problem. While current research has progressed a great deal, technology transfer to end users is limited. In this demo, we present a visualization tool for manipulating foreign language data. Using software developed for the exploration and understanding of large amounts of text data, IN-SPIRE (Hetzler & Turner 2004), we have developed a novel approach to mining and triaging large amounts of foreign language texts. By clustering documents in their native language and only using translations in the data triage phase, our system avoids the major pitfalls that plague modern machine translation. More generally, the visualization environment we have developed allows users to take advantage of current NLP technologies, including MT. We will demonstrate use of this tool to triage a corpus of foreign text.

2 IN-SPIRE

IN-SPIRE (Hetzler et al., 2004) is a visual analytics tool developed by Pacific Northwest National Laboratory to facilitate the collection and rapid understanding of large textual corpora. IN-SPIRE generates a compiled document set from mathematical signatures for each document in a set. Document signatures are clustered according to common themes to enable information retrieval and visualizations. Information is presented to the user using several visual metaphors to expose different facets of the textual data. The central visual metaphor is a galaxy view of the corpus that allows users to intuitively interact with thousands of documents, examining them by theme.

Context vectors for documents such as LSA (Deerwester et al., 1990) provide a powerful foundation for information retrieval and natural language processing techniques. IN-SPIRE leverages such representations for clustering, projection and queries-by-example (QBE). In addition to standard Boolean word queries, QBE is a process in which a user document query is converted into a mathematical signature and compared to the multi-dimensional mathematical representation of the document corpus. A spherical distance threshold adjustable by the end user controls a query result set. Using IN-SPIRE's group functionality, subsets of the corpus are identified for more detailed analyses. Information analysts can isolate meaningful document subsets into groups for hypothesis testing and the identification of trends. Depending on the corpus, one or more clusters may be less interesting to users. Removal of these documents, called "outliers", enables the investigator to more clearly understand the relationships between remaining documents. These tools expose various facets of document text and document inter-relationships.

3 Foreign Language Triage Capabilities

Information analysts need to sift through large datasets quickly and efficiently to identify relevant information for knowledge discovery. The need to sift through foreign language data complicates the task immensely. The addition of foreign language capabilities to IN-SPIRE addresses this need. We have integrated third party translators for over 40 languages and third party software for language identification. Datasets compiled with language detection allow IN-SPIRE to automatically select the most appropriate translator for each document.

To triage a foreign language dataset, the system clusters the documents in their native language

Proceedings of HLT/EMNLP 2005 Demonstration Abstracts, pages 2–3,
Vancouver, October 2005.

(with no pre-translation required). A user can then view the cluster labels, or peak terms, in the native language, or have them translated via Systran (Senellart et al., 2003) or CyberTrans (not publicly available). The user can then explore the clusters to get a general sense of the thematic coverage of the dataset. They identify clusters relevant to their interests and the tool reclusters to show more subtle themes differentiating the remaining documents. If they search for particular words, the clusters and translated labels help them distinguish the various contexts in which those words appear. Finding a cluster of document of interest, a particular document or set of documents can be viewed and translated on demand. This avoids the need to translate the entire document set, so that only the documents of interest are translated. The native text is displayed alongside the translation at all stages.

4 Evaluation

Since this is a prototype visualization tool we have yet to conduct formal user evaluations. We have begun field testing this tool with users who report successful data triage in foreign languages with which they are not familiar. We have also begun evaluations involving parallel corpora. Using Arabic English Parallel News Text (LDC 2004), which contains over 8,000 human translated documents from various Arabic new sources, we processed the English version in IN-SPIRE to view the document clusters and their labels. We also processed the Arabic version in Arabic according to the description above. The two screenshots below demonstrate that the documents clustered in similar manners (note that cluster labels have been translated in the Arabic data).

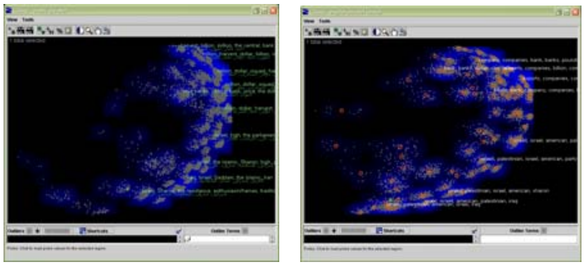


Figure 1: Galaxy view of the Arabic and English clusters and labels

To demonstrate that our clustering algorithm on the native language is an efficient and reliable method for data triage on foreign language data, we also pre-translated the data with CyberTrans and clustered on the output. Figure 3, demonstrates that similar clusters arise out of this methodology. However, the processing time was increased 15-fold with no clear advantage for data triage.

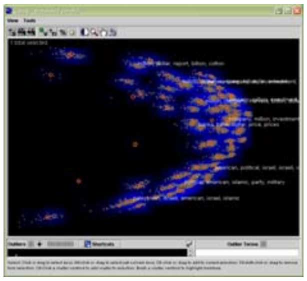


Figure 3: Galaxy view of the pre-translated Arabic to English clusters and labels

Initial user reports and comparisons with a parallel corpus demonstrate that our visualization environment enables users to search through and cluster massive amounts of data without native speaker competence or dependence on a machine translation system. Users can identify clusters of potential interest with this tool and translate (by human or machine) only those documents of relevance. We have demonstrated that this visualization tool allows users to derive high value from existing machine translation capabilities.

References

Bar-Hillel, Yehoshua, 1951. The present state of research on mechanical translation. *American Documentation* 2 (4), pp.229-237.

Hetzler, Elizabeth and Alan Turner. 2004. “Analysis Experiences Using Information Visualization,” *IEEE Computer Graphics and Applications*, 24(5):22-26.

Deerwester, S., S.T. Dumais, T.K. Landauer, G.W. Furnas, R.A. Harshman. 1990. Indexing by Latent Semantic Analysis. *Journal of the Society for Information Science*, 41(6):391-407.

Linquistic Data Consortium. 2004. http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2004T18

Senellart, Jean; Jin Yang, and Anabel Rebollo. 2003. SYSTRAN Intuitive Coding Technology. MT Summit IX. New Orleans, Louisianna.

Classummary: Introducing Discussion Summarization to Online Classrooms

Liang Zhou, Erin Shaw, Chin-Yew Lin, and Eduard Hovy
University of Southern California
Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292-6695
{liangz, shaw, hovy}@isi.edu

Abstract

This paper describes a novel summarization system, Classummary, for interactive online classroom discussions. This system is originally designed for Open Source Software (OSS) development forums. However, this new application provides valuable feedback on designing summarization systems and applying them to everyday use, in addition to the traditional natural language processing evaluation methods. In our demonstration at HLT, new users will be able to direct this summarizer themselves.

1 Introduction

The availability of many chat forums reflects the formation of globally dispersed virtual communities, one of which is the very active and growing movement of Open Source Software (OSS) development. Working together in a virtual community in non-collocated environments, OSS developers communicate and collaborate using a wide range of web-based tools including Internet Relay Chat (IRC), electronic mailing lists, and more.

Another similarly active virtual community is the distributed education community. Whether courses are held entirely online or mostly on-campus, online asynchronous discussion boards play an increasingly important role, enabling classroom-like communication and collaboration amongst students, tutors and instructors. The University of Southern California, like many other universities, employs a commercial online course management system (CMS). In an effort to bridge research and practice in education, researchers at ISI replaced the native CMS discussion board with an open source board that is currently used by selected classes. The board provides a platform for evaluating new teaching and learning technologies. Within the discussion board teachers and students post messages about course-related topics. The discussions are organized chronologically within topics and higher-level forums. These ‘live’ discussions are now enabling a new opportunity, the opportunity to apply and evaluate advanced natural language processing (NLP) technology.

Recently we designed a summarization system for technical chats and emails on the Linux kernel (Zhou and Hovy, 2005). It clusters discussions according to subtopic structures on the sub-message level, identifies immediate responding pairs using machine-learning methods, and generates subtopic-based mini-summaries for each chat log. Incorporation of this system into the ISI Discussion Board framework, called Classummary, benefits both distance learning and NLP communities. Summaries are created periodically and sent to students and teachers via their preferred medium (emails, text messages on mobiles, web, etc). This relieves users of the burden of reading through a large volume of messages before participating in a particular discussion. It also enables users to keep track of all ongoing discussions without much effort. At the same time, the discussion summarization system can be measured beyond the typical NLP evalua-

Proceedings of HLT/EMNLP 2005 Demonstration Abstracts, pages 4–5,
Vancouver, October 2005.

tion methodologies, i.e. measures on content coverage. Teachers and students' willingness and continuing interest in using the software will be a concrete acknowledgement and vindication of such research-based NLP tools. We anticipate a highly informative survey to be returned by users at the end of the service.

2 Summarization Framework

In this section, we will give a brief description of the discussion summarization framework that is applied to online classroom discussions.

One important component in the original system (Zhou and Hovy, 2005) is the sub-message clustering. The original chat logs are in-depth technical discussions that often involve multiple sub-topics, clustering is used to model this behavior. In Classummary, the discussions are presented in an organized fashion where users only respond to and comment on specific topics. Thus, it eliminates the need for clustering.

All messages in a discussion are related to the central topic, but to varying degrees. Some are answers to previously asked questions, some make suggestions and give advice where they are requested, etc. We can safely assume that for this type of conversational interactions, the goal of the participants is to seek help or advice and advance their current knowledge on various course-related subjects. This kind of interaction can be modeled as one problem-initiating message and one or more corresponding problem-solving messages, formally defined as Adjacent Pairs (AP). A support vector machine, pre-trained on lexical and structural features for OSS discussions, is used to identify the most relevant responding messages to the initial post within a topic.

Having obtained all relevant responses, we adopt the typical summarization paradigm to extract informative sentences to produce concise summaries. This component is modeled after the BE-based multi-document summarizer (Hovy et al., 2005). It consists of three steps. First, important basic elements (BEs) are identified according to their likelihood ratio (LR). BEs are automatically created minimal semantic units of the form head-modifier-relation (for example, "*Libyans | two | nn*", "*indicted | Libyans | obj*", and "*indicted | bombing | for*"). Next, each sentence is given a score which is the sum of its BE scores, computed in the first step, normalized by its length. Lastly, taking into consideration the interactions among summary sentences, a MMR (Maximum Marginal Relevancy) model (Goldstein et al., 1999) is used to extract sentences from the list of top-ranked sentences computed from the second step.

3 Accessibility

Classummary is accessible to students and teachers while classes are in session. At HLT, we will demonstrate an equivalent web-based version. Discussions are displayed on a per-topic basis; and messages belonging to a specific discussion are arranged in ascending order according to their timestamps. While viewing a new message on a topic, the user can choose to receive a summary of the discussion so far or an overall summary on the topic. Upon receiving the summary (for students, at the end of an academic term), a list of questions is presented to the user to gather comments on whether Classummary is useful. We will show the survey results from the classes (which will have concluded by then) at the conference.

References

Hovy, E., C.Y. Lin, and L. Zhou. 2005. A BE-based multi-document summarizer with sentence compression. To appear in *Proceedings of Multilingual Summarization Evaluation* (ACL 2005), Ann Arbor, MI.

Goldstein, J., M. Kantrowitz, V. Mittal, and J. Carbonell. Summarizing Text Documents: Sentence Selection and Evaluation Metrics. *Proceedings of the 22nd International ACM Conference on Research and Development in Information Retrieval* (SIGIR-99), Berkeley, CA, 121-128.

Zhou, L. and E. Hovy. 2005. Digesting virtual "geek" culture: The summarization of technical internet relay chats. To appear in *Proceedings of Association of Computational Linguistics* (ACL 2005), Ann Arbor, MI.

Demonstrating an Interactive Semantic Role Labeling System

Vasin Punyakanok Dan Roth Mark Sammons
Department of Computer Science
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA
{punyakan,danr,mssammon}@uiuc.edu

Wen-tau Yih
Microsoft Research
Redmond, WA 98052, USA
scottyih@microsoft.com

Abstract

Semantic Role Labeling (SRL) is the task of performing a shallow semantic analysis of text (i.e., *Who did What to Whom, When, Where, How*). This is a crucial step toward deeper understanding of text and has many immediate applications. Preprocessed information on text, mostly syntactic, has been shown to be important for SRL. Current research focuses on improving the performance assuming that this lower level information is given without any attention to the overall efficiency of the final system, although minimizing execution time is a necessity in order to support real world applications. The goal of our demonstration is to present an interactive SRL system that can be used both as a research and an educational tool. Its architecture is based on the state-of-the-art system (the top system in the 2005 CoNLL shared task), modified to process raw text through the addition of lower level processors, while achieving effective real time performance.

1 Introduction

Semantic parsing of sentences is believed to be an important subtask toward natural language understanding, and has immediate applications in tasks such information extraction and question answering.

We study *semantic role labeling (SRL)*, defined as follows: for each verb in a sentence, the goal is to identify all constituents that fill a semantic role, and to determine their roles (such as Agent, Patient or Instrument) and their adjuncts (such as Locative, Temporal or Manner). The PropBank project (Kingsbury and Palmer, 2002), which provides a large human-annotated corpus of semantic verb-argument relations, has opened doors for researchers to apply machine learning techniques to this task.

The focus of the research has been on improving the performance of the SRL system by using, in addition to raw text, various syntactic and semantic information, e.g. *Part of Speech (POS) tags*, *chunks*, *clauses*, *syntactic parse tree*, and *named entities*, which is found crucial to the SRL system (Punyakanok et al., 2005).

In order to support a real world application such as an interactive question-answering system, the ability of an SRL system to analyze text in real time is a necessity. However, in previous research, the overall efficiency of the SRL system has not been considered. At best, the efficiency of an SRL system may be reported in an experiment assuming that all the necessary information has already been provided, which is not realistic. A real world scenario requires the SRL system to perform all necessary preprocessing steps in real time. The overall efficiency of SRL systems that include the preprocessors is not known.

Our demonstration aims to address this issue. We present an interactive system that performs the SRL task from raw text in real time. Its architecture is based on the top system in the 2005 CoNLL shared task (Koomen et al., 2005), modified to process raw text using lower level processors but maintaining

Proceedings of HLT/EMNLP 2005 Demonstration Abstracts, pages 6–7,
Vancouver, October 2005.

good real time performance.

2 The SRL System Architecture

Our system begins preprocessing raw text by using sentence segmentation tools (available at http://l2r.cs.uiuc.edu/∼cogcomp/tools.php). Next, sentences are analyzed by a state-of-the-art syntactic parser (Charniak, 2000) the output of which provides useful information for the main SRL module.

The main SRL module consists of four stages: *pruning*, *argument identification*, *argument classification*, and *inference*. The following is the overview of these four stages. Details of them can be found in (Koomen et al., 2005).

Pruning The goal of pruning is to filter out unlikely argument candidates using simple heuristic rules. Only the constituents in the parse tree are considered as argument candidates. In addition, our system exploits a heuristic modified from that introduced by (Xue and Palmer, 2004) to filter out very unlikely constituents.

Argument Identification The argument identification stage uses binary classification to identify whether a candidate is an argument or not. We train and apply the binary classifiers on the constituents supplied by the pruning stage.

Argument Classification This stage assigns the final argument labels to the argument candidates supplied from the previous stage. A multi-class classifier is trained to classify the types of the arguments supplied by the argument identification stage.

Inference The purpose of this stage is to incorporate some prior linguistic and structural knowledge, such as "arguments do not overlap" and "each verb takes at most one argument of each type." This knowledge is used to resolve any inconsistencies in argument classification in order to generate legitimate final predictions. The process is formulated as an integer linear programming problem that takes as input confidence values for each argument type supplied by the argument classifier for each constituent, and outputs the optimal solution subject to the constraints that encode the domain knowledge.

The system in this demonstration, however, differs from its original version in several aspects. First, all syntactic information is extracted from the output of the full parser, where the original version used different information obtained from different processors. Second, the named-entity information is discarded. Finally, no combination of different parse tree outputs is performed. These alterations aim to enhance the efficiency of the system while maintaining strong performance.

Currently the system runs at the average speed of 1.25 seconds/predicate. Its performance is 77.88 and 65.87 F1-score on WSJ and Brown test sets (Carreras and Màrquez, 2005) while the original system achieves 77.11 and 65.6 on the same test sets without the combination of multiple parser outputs and 79.44 and 67.75 with the combination.

3 Goal of Demonstration

The goal of the demonstration is to present the system's ability to perform the SRL task on raw text in real time. An interactive interface allows users to input free form text and to receive the SRL analysis from our system. This demonstration can be found at http://l2r.cs.uiuc.edu/∼cogcomp/srl-demo.php.

Acknowledgments

We are grateful to Dash Optimization for the free academic use of Xpress-MP. This research is supported by ARDA's AQUAINT Program, DOI's Reflex program, and an ONR MURI Award.

References

X. Carreras and L. Màrquez. 2005. Introduction to the conll-2005 shared tasks: Semantic role labeling. In *Proc. of CoNLL-2005*.

E. Charniak. 2000. A maximum-entropy-inspired parser. In *Proc. of NAACL 2000*.

P. Kingsbury and M. Palmer. 2002. From Treebank to PropBank. In *Proc. of LREC-2002*, Spain.

P. Koomen, V. Punyakanok, D. Roth, and W. Yih. 2005. Generalized Inference with Multiple Semantic Role Labeling Systems. In *Proceedings of CoNLL-2005*.

V. Punyakanok, D. Roth, and W. Yih. 2005. The necessity of syntactic parsing for semantic role labeling. In *Proc. of IJCAI-2005*.

N. Xue and M. Palmer. 2004. Calibrating features for semantic role labeling. In *Proc. of the EMNLP-2004*.

MindNet: an automatically-created lexical resource

Lucy Vanderwende, Gary Kacmarcik, Hisami Suzuki, Arul Menezes
Microsoft Research
Redmond, WA 98052, USA
{lucyv, garykac, hisamis, arulm}@microsoft.com

Abstract

We will demonstrate MindNet, a lexical resource built automatically by processing text. We will present two forms of MindNet: as a static lexical resource, and, as a toolkit which allows MindNets to be built from arbitrary text. We will also introduce a web-based interface to MindNet lexicons (MNEX) that is intended to make the data contained within MindNets more accessible for exploration. Both English and Japanese MindNets will be shown and will be made available, through MNEX, for research purposes.

1 MindNet

A MindNet is a collection of semantic relations that is automatically extracted from text data using a broad coverage parser. Previous publications on MindNet (Suzuki et al., 2005, Richardson et al., 1998, Vanderwende 1995) have focused on the effort required to build a MindNet from the data contained in Japanese and English lexicons.

Semantic Relations

The semantic relations that are stored in MindNet are directed, labeled relationships between two words; see Table 1:

Attributive	Manner	Source
Cause	Means	Synonym
Goal	Part	Time
Hypernym	Possessor	TypicalObject
Location	Result	TypicalSubject

Table 1: A sampling of the semantic relations stored in MindNet

These semantic relations are obtained from the Logical Form analysis of our broad coverage parser NLPwin (Heidorn, 2000). The Logical Form is a labeled dependency analysis with function words removed. We have not completed an evaluation of the quality of the extracted semantic relations. Anecdotally, however, the quality varies according to the relation type, with Hypernym and grammatical relations TypicalSubject and TypicalObj being reliable, while relations such as Part and Purpose are less reliable. By making MindNet available, we solicit feedback on the utility of these labeled relationships, especially in contrast to simple co-occurrence statistics and to the heavily used hypernymy and synonymy links. Furthermore, we solicit feedback on the level of accuracy which is tolerable for specific applications.

Semantic Relation Structures

We refer to the hierarchical collection of semantic relations (*semrels*) that are automatically extracted from a source sentence as a *semrel structure*. Each semrel structure contains all of the semrels extracted from a single source sentence. A semrel structure can be viewed from the perspective of each unique word that occurs in the structure; we call these *inverted structures*. They contain the same information as the original, but with a different word placed at the root of the structure. An example semrel structure for the definition of *swallow* is given in Figure 1a, and its inversion, from the perspective of *wing* is given in Figure 1b:

```
swallow              wing
 Hyp bird              PartOf bird
        Part wing                Attrib small
        Attrib small             HypOf swallow
```

Figure 1a and b: Figure 1a is the semrel structure for the definition of swallow1, Figure 1b the inversion on wing.

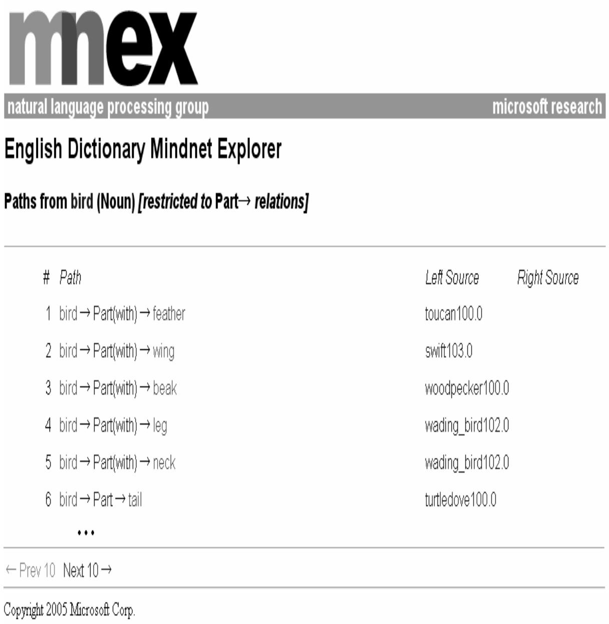
2 MNEX

MNEX (MindNet Explorer) is the web-based interface to MindNet that is designed to facilitate browsing MindNet structure and relations. MNEX displays paths based on the word or words that the

[1] *Swallow*: a small bird with wings (LDOCE). Definition abbreviated for purposes of exposition.

Proceedings of HLT/EMNLP 2005 Demonstration Abstracts, pages 8–9,
Vancouver, October 2005.

user enters. A path is a set of links that connect one word to another within either a single semrel structure or by combining fragments from multiple semrel structures. Paths are weighted for comparison (Richardson, 1997). Currently, either one or two words can be specified and we allow some restrictions to refine the path search. A user can restrict the intended part of speech of the words entered, and/or the user can restrict the paths to include only the specified relation. When two words are provided, the UI returns a list of the highest ranked paths between those two words. When only one word is given, then all paths from that word are ranked and displayed. Figure 2 shows the MNEX interface, and a query requesting all paths from the word *bird*, restricted to Noun part of speech, through the **Part** relation:



English Dictionary Mindnet Explorer

Paths from bird (Noun) *[restricted to* Part→ *relations]*

#	*Path*	*Left Source*	*Right Source*
1	bird → Part(with) → feather	toucan100.0	
2	bird → Part(with) → wing	swift103.0	
3	bird → Part(with) → beak	woodpecker100.0	
4	bird → Part(with) → leg	wading_bird102.0	
5	bird → Part(with) → neck	wading_bird102.0	
6	bird → Part → tail	turtledove100.0	
	...		

Copyright 2005 Microsoft Corp.

Figure 2: MNEX output for “bird (Noun) Part” query

3 Relation to other work

For English, WordNet is the most widely used knowledgebase. Aside from being English-only, this database was hand-coded and significant effort is required to create similar databases for different domains and languages. Projects like EuroWordNet address the monolingual aspect of WordNet, but these databases are still labor intensive to create. On the other hand, the quality of the information contained in a WordNet (Fellbaum et al., 1998) is very reliable, exactly because it was manually created. FrameNet (Baker et al., 1998) and OpenCyc are other valuable resources for English, also hand-created, that contain a rich set of relations between words and concepts. Their use is still being explored as they have been made available only recently. For Japanese, there are also concept dictionaries providing semantic relations, similarly hand-created, e.g., EDR and Nihongo Goi-taikei (NTT).

The demonstration of MindNet will highlight that this resource is automatically created, allowing domain lexical resources to be built quickly, albeit with lesser accuracy. We are confident that this is a trade-off worth making in many cases, and encourage experimentation in this area. MNEX allows the exploration of the rich set of relations through which paths connecting words are linked.

4 References

Baker, Collin F., Fillmore, Charles J., and Lowe, John B. (1998): The Berkeley FrameNet project. in Proceedings of the COLING-ACL, Montreal, Canada.

Fellbaum, C. (ed). 1998. WordNet: An Electronic Lexical Database. MIT Press.

Heidorn, G. 2000. Intelligent writing assistance. in R.Dale, H.Moisl and H.Somers (eds.), A Handbook of Natural Langauge Processing: Techniques and Applications for the Processing of Language as Text. New York: Marcel Dekker.

National Institute of Information and Communications Technology. 2001. EDR Electronic Dictionary Version 2.0 Technical Guide.

NTT Communications Science Laboratories. 1999. Goi-Taikei - A Japanese Lexicon. Iwanami Shoten.

OpenCyc. Available at: http://www.cyc.com/opencyc.

Richardson, S.D. 1997, Determining Similarity and Inferring Relations in a Lexical Knowledge Base. PhD. dissertation, City University of New York.

Richardson, S.D., W. B. Dolan, and L. Vanderwende. 1998. MindNet: Acquiring and Structuring Semantic Information from Text, In *Proceedings of ACL-COLING*. Montreal, pp. 1098-1102.

Suzuki, H., G. Kacmarcik, L. Vanderwende and A. Menezes. 2005. Mindnet and mnex. In Proceedings of the 11th Annual meeting of the Society of Natural Language Processing (in Japanese).

Vanderwende, L. 1995. Ambiguity in the acquisition of lexical information. In Proceedings of the AAAI 1995 Spring Symposium Series, symposium on representation and acquisition of lexical knowledge, 174-179.

NooJ: A Linguistic Annotation System For Corpus Processing

Max Silberztein
LASELDI
Université de Franche-Comté
Besançon, 25000 France
max.silberztein@univ-fcomte.fr

1 Introduction

NooJ is a new corpus processing system, similar to the INTEX software,[1] and designed to replace it. NooJ allows users to process large sets of texts in real time. Users can build, accumulate and manage sophisticated concordances that correspond to morphological and syntactic grammars organized in re-usable libraries.

One characteristic of NooJ is that its corpus processing engine uses large-coverage linguistic lexical and syntactic resources. This allows NooJ users to perform sophisticated queries that include any of the available morphological, lexical or syntactic properties. In comparison with INTEX, NooJ uses a new technology (.NET), a new linguistic engine, and was designed with a new range of applications in mind.

2 A new software architecture

NooJ's architecture is based on the .NET "Component programming" technology, which goes a step beyond the Object-Oriented approach (Silberztein 2004). This architecture gives it several advantages, including:

(1) it allows NooJ to read any document that can be managed on the user's computer. For instance, on a typical MS-Windows computer, NooJ can process corpora in 100+ file formats, including all variants of ASCII, ISO and Unicode, HTML, RTF, XML, MS-WORD, etc.

(2) it allows other .NET applications to access all NooJ's public methods via its software component library. For instance, a programmer can easily run a NooJ method to extract sequences of texts that match a NooJ grammar from a document that is currently opened in the current application (e.g. MS-WORD).

3 A new linguistic engine

As a corpus processing system, NooJ's most important characteristic is its linguistic engine, which is based on an annotation system. An annotation is a pair (*position, information*) that states that at a certain position in the text, a sequence is associated with a certain piece of information. NooJ processes texts that are *annotated*; annotations are stored in each text's annotation structure which is synchronized with the text buffer. Text annotations that are represented as XML tags can be easily imported to NooJ; for instance, importing the XML text:

```
<N Hum> Mr. John Smith </N>
```

will produce an annotated text in which the sequence "Mr. John Smith" is annotated with the tag "N+Hum" (annotation category "N"; property "Hum"). NooJ also provides several powerful tools to annotate texts:

-- NooJ's morphological parser is capable of analyzing complex word forms, such as Hungarian words and Germanic compounds, as well as tokenizing Asian languages. The morphological parser annotates complex word forms as sequences of annotations. For instance, the contracted word form "don't" is associated with a sequence of two annotations: <do,V+Aux+PR> and <not,ADV+Neg>.

[1] Cf. (Silberztein 1999a) for a description of the INTEX toolbox, and (Silberztein 1999b) for a description of its application as a corpus processing system. See various INTEX WEB sites for references and information on its applications, workshops and communities: http://intex.univ-fcomte.fr and the NooJ WEB site for a description of NooJ: http://www.nooj4nlp.net.

Proceedings of HLT/EMNLP 2005 Demonstration Abstracts, pages 10–11,
Vancouver, October 2005.

-- NooJ's lexical parser can process the inflection of large dictionaries for simple and compound words. For instance, the English dictionary contains 100,000+ simple words and 70,000+ compound nouns. NooJ contains large-coverage dictionaries for Arabic, Armenian, Chinese, Danish, English, French, Hungarian, Italian and Spanish. In general, running NooJ's lexical parser results in adding multiple lexical annotations to a text. The annotation system can represent all types of lexical ambiguities, such as between compounds and sequences of simple words (e.g. "round table"), overlapping or embedded compounds (e.g. "round table mat"), etc.

-- NooJ's local grammars are Recursive Transition Networks; they allow users to recognize certain sequences of texts, and to associate them with annotations. NooJ's graphical editor contains a dozen development tools to edit, test and debug local grammars, to organize them in libraries, and to apply them to texts, either as queries or to add (or filter out) annotations.

NooJ's query system and parsers can access any previously inserted annotation. For instance, the following query includes references to word forms (e.g. "mind") as well as to two annotations (written between brackets):

```
(the + these) <N+Hum> <lose>
their (mind + temper)
```

<N+Hum> matches all sequences in the text that are associated with an "N" annotation with property "Hum"; these annotations might have been added by NooJ's lexical parser (e.g. for the word "director"), or by a local grammar used to recognize human entities (e.g. for the sequence "head of this company"). Similarly, <lose> matches all sequences of the text that are associated with an annotation whose lemma is "lose"; these annotations might have been added by the lexical parser (for all conjugated forms of "to lose", e.g. "lost"), or by a local grammar that recognizes compound tenses, e.g. 'have not yet lost". When all resulting matching sequences, e.g. "These men have not yet lost their mind", have been indexed, they can be annotated, and their annotation is then instantly available either for other queries or for further cascaded parsing.

Annotated texts can be used to build complex concordances, annotate or color texts, perform a syntactic or semantic analysis, etc.

NooJ's linguistic engine, dictionaries and grammars are multilingual; that should allow users to implement translation functionalities.

4 Conclusion

Although NooJ has just come out and its technology is quite new, it is already being used by several research teams in a variety of projects. See the proceedings of the "Eight INTEX/NooJ workshop" at NooJ's WEB site: http://www.nooj4nlp.net.

5 Demo

Participants will use NooJ in order to build a named-entity recognizer from the ground up. Participants will learn how to apply a simple query to a corpus and build its corresponding concordance. Then I will demonstrate the building of a local grammar with NooJ's graphical editor, followed by a presentation of the organization of local grammars in re-usable libraries that can be shared and integrated into larger grammars.

References

Silberztein Max. 1999. INTEX: a finite-state transducer toolbox. In Theoretical Computer Science #233:1, pp. 33-46.

Silberztein Max. 1999. Indexing large corpora with INTEX. In Computer and the Humanities #33:3.

Silberztein Max, 2004. NooJ: an Object-Oriented Approach. In *INTEX pour la linguistique et le traitement automatique des langues*. C. Muller, J. Royauté, Max Silberztein eds. Cahiers de la MSH Ledoux, Presses Universitaires de Franche-Comté., pp. 359-369.

Pattern Visualization for Machine Translation Output

Adam Lopez
Institute for Advanced Computer Studies
Department of Computer Science
University of Maryland
College Park, MD 20742
alopez@cs.umd.edu

Philip Resnik
Institute for Advanced Computer Studies
Department of Linguistics
University of Maryland
College Park, MD 20742
resnik@umd.edu

Abstract

We describe a method for identifying systematic patterns in translation data using part-of-speech tag sequences. We incorporate this analysis into a diagnostic tool intended for developers of machine translation systems, and demonstrate how our application can be used by developers to explore patterns in machine translation output.

1 Introduction

Over the last few years, several automatic metrics for machine translation (MT) evaluation have been introduced, largely to reduce the human cost of iterative system evaluation during the development cycle (Papineni et al., 2002; Melamed et al., 2003). All are predicated on the concept of *n*-gram matching between the sentence hypothesized by the translation system and one or more *reference translations*—that is, human translations for the test sentence. Although the formulae underlying these metrics vary, each produces a single number representing the “goodness” of the MT system output over a set of reference documents. We can compare the numbers of competing systems to get a coarse estimate of their relative performance. However, this comparison is holistic. It provides no insight into the specific competencies or weaknesses of either system.

Ideally, we would like to use automatic methods to provide immediate diagnostic information about the translation output—*what* the system does well, and what it does poorly. At the most general level, we want to know how our system performs on the two most basic problems in translation – word translation and reordering. Holistic metrics are at odds with day-to-day hypothesis testing on these two problems. For instance, during the development of a new MT system we may may wish to compare competing reordering models. We can incorporate each model into the system in turn, and rank the results on a test corpus using BLEU (Papineni et al., 2002). We might then conclude that the model used in the highest-scoring system is best. However, this is merely an implicit test of the hypothesis; it does not tell us anything about the specific strengths and weaknesses of each method, which may be different from our expectations. Furthermore, if we understand the relative strengths of each method, we may be able to devise good ways to combine them, rather than simply using the best one, or combining strictly by trial and error. In order to fine-tune MT systems, we need fine-grained error analysis.

What we would really like to know is how well the system is able to capture systematic reordering patterns in the input, which ones it is successful with, and which ones it has difficulty with. Word *n*-grams are little help here: they are too many, too sparse, and it is difficult to discern general patterns from them.

2 Part-of-Speech Sequence Recall

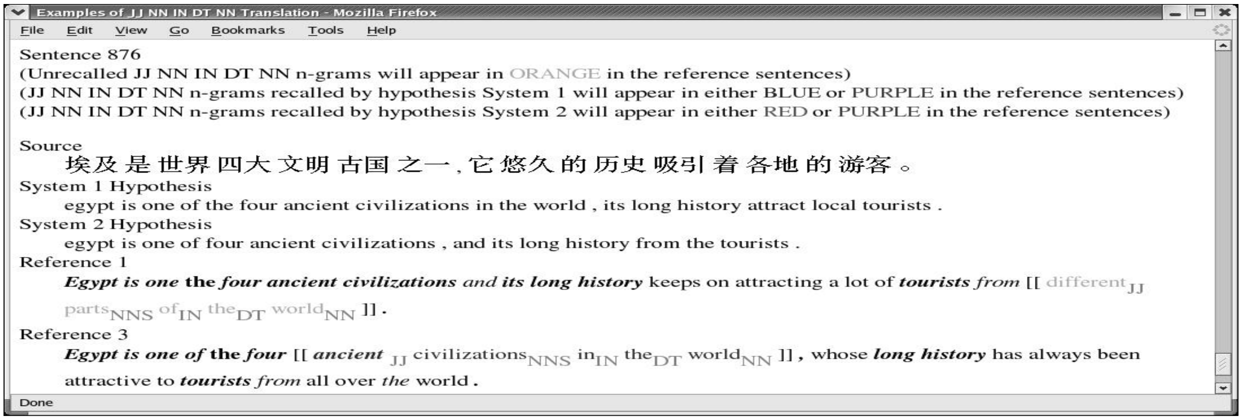
In developing a new analysis method, we are motivated in part by recent studies suggesting that word reorderings follow general patterns with respect to syntax, although there remains a high degree of flexibility (Fox, 2002; Hwa et al., 2002). This suggests that in a comparative analysis of two MT systems (or two versions of the same system), it may be useful to look for syntactic patterns that one system (or version) captures well in the target language and the other does not, using a syntax-based, recall-oriented metric.

As an initial step, we would like to summarize reordering patterns using part-of-speech sequences. Unfortunately, recent work has confirmed the intuition that applying statistical analyzers trained on well-formed text to the noisy output of MT systems produces unuseable results (e.g. (Och et al., 2004)). Therefore, we make the conservative choice to apply annotation only to the reference corpus. Word *n*-gram correspondences with a reference translation are used to infer the part-of-speech tags for words in the system output.

The method:

1. Part-of-speech tag the reference corpus. We used

Proceedings of HLT/EMNLP 2005 Demonstration Abstracts, pages 12–13,
Vancouver, October 2005.



Sentence 876
(Unrecalled JJ NN IN DT NN n-grams will appear in ORANGE in the reference sentences)
(JJ NN IN DT NN n-grams recalled by hypothesis System 1 will appear in either BLUE or PURPLE in the reference sentences)
(JJ NN IN DT NN n-grams recalled by hypothesis System 2 will appear in either RED or PURPLE in the reference sentences)

Source
埃及 是 世界 四大 文明 古国 之一 , 它 悠久 的 历史 吸引 着 各地 的 游客 。
System 1 Hypothesis
egypt is one of the four ancient civilizations in the world , its long history attract local tourists .
System 2 Hypothesis
egypt is one of four ancient civilizations , and its long history from the tourists .
Reference 1
Egypt is one **the** ***four ancient civilizations*** *and* ***its long history*** keeps on attracting a lot of ***tourists*** *from* [[different$_{JJ}$ parts$_{NNS}$ of$_{IN}$ the$_{DT}$ world$_{NN}$]] .
Reference 3
Egypt is one of **the** ***four*** [[***ancient*** $_{JJ}$ civilizations$_{NNS}$ in$_{IN}$ the$_{DT}$ world$_{NN}$]] , whose ***long history*** has always been attractive to ***tourists*** *from* all over *the* world .

Figure 1: Comparing two systems that differ significantly in their recall for POS n-gram JJ NN IN DT NN. The interface uses color to make examples easy to find.

MXPOST (Ratnaparkhi, 1996), and in order to discover more general patterns, we map the tag set down after tagging, e.g. NN, NNP, NNPS and NNS all map to NN.

2. Compute the frequency $\mathit{freq}(t_i \ldots t_j)$ of every possible tag sequence $t_i \ldots t_j$ in the reference corpus.
3. Compute the correspondence between each hypothesis sentence and *each* of its corresponding reference sentences using an approximation to maximum matching (Melamed et al., 2003). This algorithm provides a list of *runs* or contiguous sequences of words $e_i \ldots e_j$ in the reference that are also present in the hypothesis. (Note that runs are order-sensitive.)
4. For each recalled *n*-gram $e_i \ldots e_j$, look up the associated tag sequence $t_i \ldots t_j$ and increment a counter $\mathit{recalled}(t_i \ldots t_j)$

Using this method, we compute the recall of tag patterns, $R(t_i \ldots t_j) = \mathit{recalled}(t_i \ldots t_j)/\mathit{freq}(t_i \ldots t_j)$, for all patterns in the corpus.

To compare two systems (which could include two versions of the same system), we identify POS n-grams that are recalled significantly more frequently by one system than the other, using a difference-of-proportions test to assess statistical significance. We have used this method to analyze the output of two different statistical machine translation models (Chiang et al., 2005).

3 Visualization

Our demonstration system uses an HTML interface to summarize the observed pattern recall. Based on frequent or significantly-different recall, the user can select and visually inspect color-coded examples of each pattern of interest in context with both source and reference sentences. An example visualization is shown in Figure 1.

4 Acknowledgements

The authors would like to thank David Chiang, Christof Monz, and Michael Subotin for helpful commentary on this work. This research was supported in part by ONR MURI Contract FCPO.810548265 and Department of Defense contract RD-02-5700.

References

David Chiang, Adam Lopez, Nitin Madnani, Christof Monz, Philip Resnik, and Michael Subotin. 2005. The hiero machine translation system: Extensions, evaluation, and analysis. In *Proceedings of HLT/EMNLP 2005*, Oct.

Heidi J. Fox. 2002. Phrasal cohesion and statistical machine translation. In *Proceedings of the 2002 Conference on EMNLP*, pages 304–311, Jul.

Rebecca Hwa, Philip Resnik, Amy Weinberg, and Okan Kolak. 2002. Evaluating translational correspondence using annotation projection. In *Proceedings of the 40th Annual Meeting of the ACL*, pages 392–399, Jul.

I. Dan Melamed, Ryan Green, and Joseph P. Turian. 2003. Precision and recall of machine translation. In *HLT-NAACL 2003 Companion Volume*, pages 61–63, May.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. A smorgasbord of features for statistical machine translation. In *Proceedings of HLT-NAACL 2004*, pages 161–168, May.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the ACL*, pages 311–318, Jul.

Adwait Ratnaparkhi. 1996. A maximum entropy model for part-of-speech tagging. In *Proceedings of the Conference on EMNLP*, pages 133–142, May.

Prague Dependency Treebank as an exercise book of Czech

Barbora Hladká and **Ondřej Kučera**
Institute of Formal and Applied Linguistics
Charles University
Malostranské nám. 25
118 00 Prague, Czech Republic
hladka@ufal.mff.cuni.cz, ondrej.kucera@centrum.cz

Abstract

There was simply linguistics at the beginning. During the years, linguistics has been accompanied by various attributes. For example *corpus* one. While a name corpus is relatively young in linguistics, its content related to a language - collection of texts and speeches - is nothing new at all. Speaking about corpus linguistics nowadays, we keep in mind collecting of language resources in an electronic form. There is one more attribute that computers together with mathematics bring into linguistics - *computational*. The progress from working with corpus towards the computational approach is determined by the fact that electronic data with the ”unlimited” computer potential give opportunities to solve natural language processing issues in a fast way (with regard to the possibilities of human being) on a statistically significant amount of data.

Listing the attributes, we have to stop for a while by the notion of *annotated* corpora. Let us build a big corpus including all Czech text data available in an electronic form and look at it as a sequence of characters with the space having dominating status – a separator of words. It is very easy to compare two words (as strings), to calculate how many times these two words appear next to each other in a corpus, how many times they appear separately and so on. Even more, it is possible to do it for every language (more or less). This kind of calculations is language independent – it is not restricted by the knowledge of language, its morphology, its syntax. However, if we want to solve more complex language tasks such as machine translation we cannot do it without deep knowledge of language. Thus, we have to transform language knowledge into an electronic form as well, i.e. we have to formalize it and then assign it to words (e.g., in case of morphology), or to sentences (e.g., in case of syntax). A corpus with additional information is called an annotated corpus.

We are lucky. There is a real annotated corpus of Czech – Prague Dependency Treebank (PDT). PDT belongs to the top of the world corpus linguistics and its second edition is ready to be officially published (for the first release see (Hajič et al., 2001)). PDT was born in *Prague* and had arisen from the tradition of the successful Prague School of Linguistics. The *dependency* approach to a syntactical analysis with the main role of verb has been applied. The annotations go from the morphological level to the tectogrammatical level (level of underlying syntactic structure) through the intermediate syntactical-analytical level. The data (2 mil. words) have been annotated in the same direction, i.e., from a more simple level to a more

Proceedings of HLT/EMNLP 2005 Demonstration Abstracts, pages 14–15,
Vancouver, October 2005.

complex one. This fact corresponds to the amount of data annotated on a particular level. The largest number of words have been annotated morphologically (2 mil. words) and the lowest number of words tectogramatically (0.8 mil. words). In other words, 0.8 million words have been annotated on all three levels, 1.5 mil. words on both morphological and syntactical level and 2 mil. words on the lowest morphological level.

Besides the verification of 'pre-PDT' theories and formulation of new ones, PDT serves as training data for machine learning methods. Here, we present a system **Styx** that is designed to be an exercise book of Czech morphology and syntax with exercises directly selected from PDT. The schoolchildren can use a computer to write, to draw, to play games, to page encyclopedia, to compose music - why they could not use it to parse a sentence, to determine gender, number, case, ... ? While the Styx development, two main phases have been passed:

1. **transformation** of an academic version of PDT into a school one. 20 thousand sentences were automatically selected out of 80 thousand sentences morphologically and syntactically annotated. The complexity of selected sentences exactly corresponds to the complexity of sentences exercised in the current textbooks of Czech. A syntactically annotated sentence in PDT is represented as a tree with the same number of nodes as is the number of the words in the given sentence. It differs from the schemes used at schools (Grepl and Karlík, 1998). On the other side, the linear structure of PDT morphological annotations was taken as it is – only morphological categories relevant to school syllabuses were preserved.
2. **proposal** and **implementation of exercises**. The general computer facilities of basic and secondary schools were taken into account while choosing a potential programming language to use. The Styx is implemented in Java that meets our main requirements – platform-independent system and system stability.

At least to our knowledge, there is no such system for any language corpus that makes the schoolchildren familiar with an academic product. At the same time, our system represents a challenge and an opportunity for the academicians to popularize a field devoted to the natural language processing with promising future.

A number of electronic exercises of Czech morphology and syntax were created. However, they were built manually, i.e. authors selected sentences either from their minds or randomly from books, newspapers. Then they analyzed them manually. In a given manner, there is no chance to build an exercise system that reflects a real usage of language in such amount the Styx system fully offers.

References

Jan Hajič, Eva Hajičová, Barbora Hladká, Petr Pajas, Jarmila Panevová, and Petr Sgall. 2001. *Prague Dependency Treebank 1.0 (Final Production Label)* CD-ROM, CAT: LDC2001T10, ISBN 1-58563-212-0, Linguistic Data Consortium.

Miroslav Grepl and Petr Karlík 1998. *Skladba češiny. [Czech Langauge.]* Votobia, Praha.

Translation Exercise Assistant:
Automated Generation of Translation Exercises for Native-Arabic Speakers Learning English

Jill Burstein
Educational Testing Service
Princeton, NJ 08541
jburstein@ets.org

Daniel Marcu
Language Weaver, Inc
Marina del Rey, CA 90292
dmarcu@languageweaver.com

1. Introduction

Machine translation has clearly entered into the marketplace as a helpful technology. Commercial applications are used on the internet for automatic translation of web pages and news articles. In the business environment, companies offer software that performs automatic translations of web sites for localization purposes, and translations of business documents (e.g., memo and e-mails). With regard to education, research using machine translation for language learning tools has been of interest since the early 1990's (Anderson, 1993, Richmond, 1994, and Yasuda, 2004), though little has been developed. Very recently, Microsoft introduced a product called *Writing Wizard* that uses machine translation to assist with business writing for native Chinese speakers. To our knowledge, this is currently the only deployed education-based tool that uses machine translation.

Currently, all writing-based English language learning (ELL) writing-based products and services at Educational Testing Service rely on e-rater automated essay scoring and the *Critique* writing analysis tool capabilities (Burstein, Chodorow, and Leacock, 2004). In trying to build on a portfolio of innovative products and services, we have explored using machine translation toward the development of new ELL-based capabilities. We have developed a prototype system for automatically generating translation exercises in Arabic --- the *Translation Exercise Assistant*.

Translation exercises are one kind of task that teachers can offer to give students practice with specific grammatical structures in English. Our hypothesis is that teachers could use such a tool to help them create exercises for the classroom, homework, or quizzes. The idea behind our prototype is a capability that can be used either by classroom teachers to help them generate sentence-based translation exercises from an infinite number of Arabic language texts of their choice. The capability might be integrated into a larger English language learning application. In this latter application, these translation exercises could be created by classroom teachers for the class or for individuals who may need extra help with particular grammatical structures in English. Another potential use of this system that has been discussed is to use it in ESL classrooms in the United States, to allow teachers to offer exercises in students' native language, especially for students who are competent in their own language, but only beginners in English.

We had two primary goals in mind in developing our prototype. First, we wanted to evaluate how well the machine translation capability itself would work with this application. In other words, how useful were the system outputs that are based on the machine translations? We also wanted to know to what extent this kind of tool facilitated the task of creating translation exercise items. So, how much time is involved for a teacher to manually create these kinds of items versus using the exercise assistant tool to create them? Manually creating such an item involves searching through numerous reference sources (e.g., paper or web-based version of newspapers), finding sentences with the relevant grammatical structure in the source language (Arabic), and then manually producing an English translation that can be used as an answer key.

To evaluate these aspects, we implemented a graphical user interface that offered our two users the ability to create sets of translation

Proceedings of HLT/EMNLP 2005 Demonstration Abstracts, pages 16–17,
Vancouver, October 2005.

exercise items for six pre-selected, grammatical structures. For each structure the system automatically identified and offered a set of 200 system-selected potential sentences per category. For the exercise creation task, we collected timing information that told us how long it took users to create 3 exercises of 10 sentences each, for each category. In addition, users rated a set of up to 200 Arabic sentences with regard to if they were usable as translation exercise items, so that we could gauge the proportion of sentences selected by the application. These were the sentences that remained in the set of 200 because they were not selected for an exercise. Two teachers participated in the evaluation of our prototype. One of the users also did the task manually.

2. Translation Exercise Selection

2.1 Data Sets

The source of the data was Arabic English Parallel News Part 1 and the Multiple Translation Arabic Part 1 corpus from the Linguistic Data Consortium.[1] Across these data sets we had access to about 45,000 Arabic sentences from Arabic journalistic texts taken from Ummah Press Service, Xinhua News and the AFP News Service available for this research. We used approximately 10,000 of these Arabic sentences for system development, and selected sentences from the remaining Arabic sentences for use with the interface.[2]

2.2 System Description

We used Language Weaver's[3] Arabic-to-English system to translate the Arabic sentences in the data sets. We built a module to find the relevant grammatical structures in the English translations. This module first passes the English translation to a part-of-speech tagger that assigns a part-of-speech to each word in the sentence. Another module identifies regular expressions for the relevant part-of-speech sequences in the sentences, corresponding to one of these six grammatical structures: a) *subject-verb agreemen*t, b) *complex verbs*, c) *phrasal verb*s, d) *nominal compounds*, e) *prepositions*, and f) *adjective modifier phrases*. When the appropriate pattern was found in the English translation, the well-formed Arabic sentence that corresponds to that translation is added to the set of potential translation exercise sentence candidates in the interface.

2.3 Results

The outcome of the evaluation indicated that between 98% and 100% of automatically-generated sentence-based translation items were selected by both users as usable for translation items. In addition, the time involved to create the exercises using the tool was 2.6 times faster than doing the task manually.

References

Anderson, Don D. (1995) "Machine Translation as a Tool in Second Language Learning", *CALICO Journal* 13.1, 68–97.

Burstein, J., Chodorow, M., & Leacock, C. (2004). Automated essay evaluation: The Criterion online writing service. *AI Magazine, 25*(3), 27-36.

Johnson, Rod (1993) "MT Technology and Computer-Aided Language Learning", in Sergei Nirenburg (ed.) Progress in Machine Translation, Amsterdam: IOS and Tokyo: Ohmsha, pages 286–287.

Richmond, Ian M. (1994) "Doing it backwards: Using translation software to teach target-language grammaticality", Computer Assisted Language Learning 7, 65–78.

Yasuda, K. Sugaya F., Sumita E, Takezawa T., Kikui G., Yamamoto, S. (2004). Automatic Measuring of English Language Proficiency using MT Evaluation Technology. Proceedings of e-Learning workshop, COLING 2004, Geneva, Switzerland.

[1] The LDC reference numbers for these corpora are: LDC2004T18 and LDC2003T18.

[2] To avoid producing sentences with overly complicated structures, we applied two constraints to the English translation: 1) it contained 20 words or less, and 2) it contained only a single sentence.

[3] See http://www.languageweaver.com.

WebExperimenter for multiple-choice question generation

Ayako Hoshino
Interfaculty Initiative in Information Studies
University of Tokyo
7-3-1 Hongo, Bunkyo, Tokyo,
113-0033, JAPAN

Hiroshi Nakagawa
Information Technology Center
University of Tokyo
7-3-1 Hongo, Bunkyo, Tokyo,
113-0033, JAPAN

{hoshino,nakagawa}@dl.itc.u-tokyo.ac.jp

1 Aim

Automatic generation of multiple-choice questions is an emerging topic in application of natural language processing. Particularly, applying it to language testing has been proved to be useful (Sumita et al., 2005).

This demo presents an novel approach of question generation using machine learning we have introduced in (Hoshino and Nakagawa, 2005). Our study aims to generate TOEIC-like [1] multiple choice, fill-in-the-blank questions from given text using a classifier trained on a set of human-made questions. The system comprises of **a question pool**, which is a database of questions, **an instance converter** which does feature extraction, etc. for machine learning and **a question generator**. Each step of learning and generation is conducted through a web-browser.

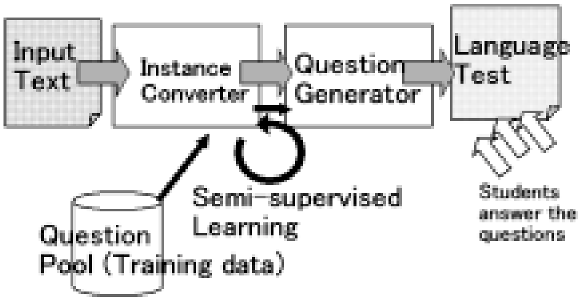


Figure 1: A system diagram

The demo serves for the following three purposes; To facilitates repeating the experiment with different parameters, to demonstrate our method of question generation by showing the result of each steps, and to collect the data (training data and the students' answers) from multiple users in possibly different places.

2 Processes

An experiment is performed in a sequence of processes in each of which the system allows the user to change input/parameters and shows the result. The demo follows the processes described in the following.

Input Questions
The questions in the question pool are listed on the browser. The user can modify those questions or add new ones.

Convert to Instances
Each question in the question pool is automatically converted into instances each of which represents a possible blank position.

A sentence is [ ] to instances.
1.convert 2. converted 3. converts 4. conversion

Above question sentence is converted into the following instances, then, features such as POS [2], lemma, POS of the previous word, POS of the next word, position-in-sentence, sentence length are assigned to each instance in a totally automatic fashion.

We decide a blank position for a question by classifying an instance into *true* or *false*. Temporally,

[1] TOEIC: Test of English for International Communication

[2] Part-of-speech tags are tagged by a modified version of the Tree Tagger by the University of Stuttgart.

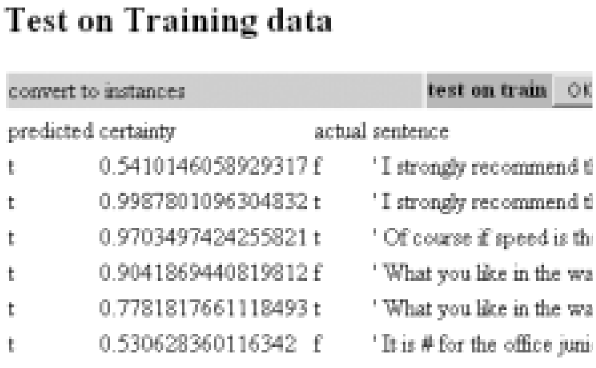
Proceedings of HLT/EMNLP 2005 Demonstration Abstracts, pages 18–19,
Vancouver, October 2005.

the original blank positions are labeled *true*, and the shifted ones are labeled as *false*.

false	[ ] sentence is converted to multiple instances.
false	A [ ] is converted to multiple instances.
false	A sentence [ ] converted to multiple instances.
true	A sentence is [ ] to multiple instances.
false	A sentence is converted [ ] multiple instances.
false	A sentence is converted to [ ] instances.
false	A sentence is converted to multiple [ ] .
false	A sentence is converted to multiple instances [ ]

First Training
The instances are fed to a classifier selected among ones of Naive Bayes, K-Nearest Neighbors, Logistic Regression.

Test on Train
A semi-supervised learning is conducted here for the purpose of discovering falsely labeled *true* instances (which correspond with blank positions shifted from the original ones, but has the same properties with *true* instances) and the labels of those instances are changed. The classifier is re-trained on the data with new labels. This process can be iterated several times.



Test on Training data

```
predicted certainty          actual sentence
t         0.5410146058929317 f      'I strongly recommend t
t         0.9987801096304832 t      'I strongly recommend t
t         0.9703497424255821 t      'Of course if speed is th
t         0.9041869440819812 f      'What you like in the wa
t         0.7781817661118493 t      'What you like in the wa
t         0.530628360116342  f      'It is # for the office jun
```

Figure 2: A screenshot of a result of test on train

The instances classified as *true* are shown along with its temporal label and its certainty value (certainty for an instance to belong to a class *true*) given by the classifier.

Supply Test Data
The user supplies a source text for question generation from a text area. The test data is converted into instances in the same way as the training data.

Classify Test
The test instances are classified by the classifier which has been trained through semi-supervised learning. *True* instances which represents blank position are shown. Instances with a label *true* are passed to the next step of deciding distractors, where instances with *false* are discarded.

Generate Questions
A set of wrong answers (called *distractors*) are decided. The user can choose a method of deciding distractors among WordNet, Edit Distance, Mutual Information and Random. The resulting four-choice questions are shown.

Question Session
An interface to collect the students' answers to generated questions is scheduled. The students' performance is used to evaluate the questions.

3 Related Studies

The application of NLP techniques to generation of multiple-choice questions does not have a long history. Few attempts had been made before (Mitkov and Ha, 2003), in which a semi-automatic question generation on student's knowledge of linguistic terms are evaluated. Sumita et al. used automatically generated questions to measure test taker's proficiency in English (2005). We are proposing a machine learning approach which depends on a training on a collection of manually made questions (Hoshino and Nakagawa, 2005).

References

Ayako Hoshino and Hiroshi Nakagawa. 2005. A real-time multiple-choice question generation for language testing: A preliminary study. In *Proceedings of the ACL 2005 The Second Workshop on Building Educational Applications Using Natural Language Processing*, to appear.

Ruslan Mitkov and Le An Ha. 2003. Computer-aided generation of multiple-choice tests. In *Proceedings of the HLT-NAACL 2003 Workshop on Building Educational Applications Using Natural Language Processing*, pages 17 – 22, Edmonton, Canada, May.

Eiichiro Sumita, Fumiaki Sugaya, and Seiichi Yamamoto. 2005. Measuring non-native speaker's proficiency of english by using a test with automatically-generated fill-in-the-blank questions. In *Proceedings of the ACL 2005 The Second Workshop on Building Educational Applications Using Natural Language Processing*, to appear.

DialogueView: an Annotation Tool for Dialogue

Fan Yang and **Peter A. Heeman**
Center for Spoken Langauge Understanding
OGI School of Science & Engineering
Oregon Health & Science University
20000 NW Walker Rd., Beaverton OR, U.S.A. 97006
{fly, heeman}@cslu.ogi.edu

1 Introduction

There is growing interest in collecting and annotating corpora of language use. Annotated corpora are useful for formulating and verifying theories of language interaction, and for building statistical models to allow a computer to naturally interact with people.

A lot of annotation tools have been built or are being built. CSLU Toolkit (Sutton et al., 1998) and Emu (Cassidy and Harrington, 2001) are built for words transcription or speech events (such as accent); DAT is built for coding dialogue acts using the DAMSL scheme (Core and Allen, 1997); Nb is built for annotating hierarchical discourse structure (Flammia, 1998); annotation toolkits, such as Mate (McKelvie et al., 2001), AGTK (Bird et al., 2001), and Nite (Carletta et al., 2003), are built for users to create their own tools. In this demo, we will present a novel tool, DialogueView, for annotating speech repairs, utterance boundaries, utterance tags, and hierarchical discourse structure altogether.

The annotation tool, DialogueView, consists of three views: WordView, UtteranceView, and BlockView. These three views present different abstractions of a dialogue, which helps users better understand what is happening in the dialogue. WordView shows the words time-aligned with the audio signal. UtteranceView shows the dialogue as a sequence of utterances. It abstracts away from the exact timing of the words and can even skip words, based on WordView annotations, that do not impact the progression of the dialogue. BlockView shows the dialogue as a hierarchy of discourse blocks, and abstracts away from the exact utterances that were said. Annotations are done at the view that is most appropriate for what is being annotated. The tool allows users to easily navigate among the three views and it automatically updates all views when changes are made in one view.

DialogueView makes use of multiple views to present different abstractions of a dialogue to users. Abstraction helps users focus on what is important for different annotation tasks. For example, for annotating speech repairs, utterance boundaries, and overlapping and abandoned utterances, WordView provides the exact timing information. For coding speech act tags and hierarchical discourse structure, UtteranceView shows a broader context and hides such low-level details.

In this presentation, we will show how DialogueView helps users annotate speech repairs, utterance boundaries, utterance tags, and hierarchical discourse blocks. Researchers studying dialogue might want to use this tool for annotating these aspects of their own dialogues. We will also show how the idea of abstraction in DialogueView helps users understand and annotate a dialogue. Although DialogueView focuses on spoken dialogue, we feel that abstraction can be used in annotating monologues, multi-party, and multi-modal interaction, with any type of annotations, such as syntactic structure, semantics and co-reference. Researchers might want to adopt the use of abstraction in their own annotation tools.

2 WordView

The first view is *WordView*, which takes as input two audio files (one for each speaker), the words said by each speaker and the start and stop times of each word (in XML format), and shows the words time-aligned with the audio signal. This view is ideal for seeing the exact timing of speech, especially overlapping speech. Users can annotate speech repairs, utterance boundaries, and utterance tags in WordView.

WordView gives users the ability to select a region of the dialogue and to play it. Users can play each speaker channel individually or both combined. Furthermore, DialogueView allows users to aurally verify their speech repair annotations. WordView supports playing a region of speech but with the annotated reparanda and editing terms skipped over. We have found this useful in deciding whether a speech repair is correctly annotated. If one has annotated the repair correctly, the edited speech will sound fairly natural.

3 UtteranceView

The annotations in WordView are utilized in building the next view, *UtteranceView*. This view shows the utterances of two speakers as if it were a script for a movie. To derive a single ordering of the utterances of the two

Proceedings of HLT/EMNLP 2005 Demonstration Abstracts, pages 20–21,
Vancouver, October 2005.

speakers, we use the start time of each utterance as annotated in WordView. We refer to this process as *linearizing* the dialogue (Heeman and Allen, 1995). The order of the utterances should show how the speakers are sequentially adding to the dialogue, and is our motivation for defining utterances as being small enough so that they are not affected by subsequent speech of the other speaker.

Users can annotate utterance tags in UtteranceView besides WordView. WordView is more suitable for tags that depend on the exact timing of the words, or a very local context, such as whether an utterance is abandoned or incomplete, or whether there is overlap speech. UtteranceView is more suitable for tags that relate the utterance to other utterances in the dialogue, such as whether an utterance is an answer, a statement, a question, or an acknowledgment. Whether an annotation tag can be used in WordView or UtteranceView (or both) is specified in the configuration file. Which view a tag is used in does not affect how it is stored in the annotation files (also in XML format).

In UtteranceView, users can annotate hierarchical groupings of utterances. We call each grouping a *block*, and blocks can have other blocks embedded inside of them. Each block is associated with a *summary*, which users need to fill in. Blocks can be closed; when a block is closed, it is replaced by its summary, which is displayed as if it were said by the speaker who initiated the block. Just as utterances can be tagged, so can discourse blocks. The block tags scheme is also specified in the configuration file.

UtteranceView supports two types of playback. The first playback simply plays both channels mixed, which is exactly what is recorded. The second playback is slightly different. It takes the linearization into account and dynamically builds an audio file in which each utterance in turn is concatenated together, and a 0.5 second pause is inserted between each utterance. This gives the user an idealized rendition of the utterances, with overlapping speech separated. By comparing these two types of playbacks, users can aurally check if their linearization of the dialogue is correct.

Users can use the configuration file to customize UtteranceView. Typically, UtteranceView gives users a *clean display* of what is going on in a dialogue. This clean display removes reparanda and editing terms in speech repairs, and it also removes abandoned speech, which has no contributions to the conversation.[1] UtteranceView also supports adding texts or symbols to an utterance based on the tags, such as adding "?" after a question, "..." after an incomplete utterance, and "+" at both the beginning and end of an overlapping utterance to signal the overlap. (c.f. *Childes* scheme (MacWhinney, 2000)).

4 BlockView

In addition to WordView and UtteranceView, we are experimenting with a third view, which we call *BlockView*. This view shows the hierarchical structure of the discourse by displaying the summary and intention (DSP) for each block, indented appropriately. BlockView gives a very concise view of the dialogue. It is also convenient for navigating in the dialogue. By highlighting a line and then pressing *Sync*, the user can see the corresponding part of the dialogue in UtteranceView and WordView.

5 Availability

DialogueView is written in Incr Tcl/Tk. We also use the snack package for audio support; hence DialogueView supports audio file formats of WAV, MP3, AU, and others (see *http://www.speech.kth.se/snack/* for the complete list). DialogueView has been tested on Microsoft Windows (2000 and XP) and Redhat Enterprise Linux.

DialogueView is freely available for research and educational use. Users should first install a standard distribution of Tcl/Tk, such as ActiveTcl from *http://www.tcl.tk*, and then download DialogueView from *http://www.cslu.ogi.edu/DialogueView*. The distribution also includes some examples of annotated dialogues.

References

Steven Bird et al. 2001. Annotation tools based on the annotation graph API. In *Proceedings of ACL/EACL 2001 Workshop on Sharing Tools and Resources for Research and Education*.

Jean Carletta et al. 2003. The NITE XML toolkit: flexible annotation for multi-modal language data. *Behavior Research Methods, Instruments, and Computers*, April. Special Issue on Measuring Behavior.

Steve Cassidy and Jonathan Harrington. 2001. Multi-level annotation in the Emu speech database management system. *Speech Communication*, 33:61–77.

Mark G. Core and James F. Allen. 1997. Coding dialogues with the DAMSL annotation scheme. In *Proceedings of AAAI Fall 1997 Symposium*.

Giovanni Flammia. 1998. *Discourse Segmentation Of Spoken Dialogue: An Empirical Approach*. Ph.D. thesis, Massachusetts Institute of Technology.

Peter A. Heeman and James Allen. 1995. Dialogue transcription tools. Trains Technical Note 94-1, URCS, March.

Brian MacWhinney. 2000. *The CHILDES Project: Tools for Analyzing Talk*. Mahwah, NJ:Lawrence Erlbaum Associates, third edition.

D. McKelvie, et al. 2001. The MATE Workbench - An annotation tool for XML coded speech corpora. *Speech Communication*, 33(1-2):97–112. Special issue, ”speech Annotation and Corpus Tools”.

Stephen Sutton et al.. 1998. Universal speech tools: The CSLU toolkit. In *Proceedings of 5th ICSLP*, Australia.

[1] Note that these clean processes are optional. Users can specify them in the configuration file.

Extracting Information about Outbreaks of Infectious Epidemics

Roman Yangarber **Lauri Jokipii**
Department of Computer Science
University of Helsinki, Finland
first.last@cs.helsinki.fi

Antti Rauramo
Index, Oy
Helsinki, Finland

Silja Huttunen
Department of Linguistics
University of Helsinki, Finland

Abstract

This work demonstrates the ProMED-PLUS Epidemiological Fact Base. The facts are automatically extracted from plain-text reports about outbreaks of infectious epidemics around the world. The system collects new reports, extracts new facts, and updates the database, in real time. The extracted database is available on-line through a Web server.

1 Introduction

Information Extraction (IE) is a technology for finding facts in plain text, and coding them in a logical representation, such as a relational database.

Much published work on IE reports on "closed" experiments; systems are built and evaluated based on carefully annotated corpora, at most a few hundred documents.[1] The goal of the work presented here is to explore the IE process *in the large*: the system integrates a number of off-line and on-line components around the core IE engine, and serves as a base for research on a wide range of problems.

The system is applied to a large dynamic collection of documents in the epidemiological domain, containing tens of thousands of documents. The topic is outbreaks of infectious epidemics, affecting humans, animals and plants. To our knowledge, this is the first large-scale IE database in the epidemiological domain publicly accessible on-line.[2]

2 System Description

The architecture of the ProMED-PLUS system[3] is shown in Fig. 1. The core IE Engine (center) is implemented as a sequence, or "pipeline," of stages:

- Layout analysis, tokenisation, lexical analysis;
- Name recognition and classification;
- Shallow syntactic analysis;
- Resolution of co-reference among entities;
- Pattern-based event matching and role mapping;
- Normalisation and output generation

The database (DB) contains facts extracted from ProMED-Mail, a mailing list about epidemic outbreaks.[4]

The IE engine is based in part on earlier work, (Grishman et al., 2003). Novel components use machine learning at several stages to enhance the performance of the system and the quality of the extracted data: acquisition of domain knowledge for populating the knowledge bases (left side in Fig. 1), and automatic post-validation of extracted facts for detecting and reducing errors (upper right). Novel features include the notion of confidence,[5] and aggregation of separate facts into outbreaks across multiple reports, based on confidence.

Operating in the large is essential, because the learning components in the system rely on the availability of large amounts of data. Knowledge

[1] Cf., e.g., the MUC and ACE IE evaluation programmes.

[2] On-line IE databases do exist, e.g., CiteSeer, but none that extract multi-argument events from plain natural-language text.

[3] **PLUS**: Pattern-based Learning and Understanding System.

[4] ProMED, www.promedmail.org, is the Program for Monitoring Emerging Diseases, of the International Society for Infectious Diseases. It is one of the most comprehensive sources of reports about the spread of infectious epidemics around the world, collected for over 10 years.

[5] Confidence for individual fields of extracted facts, and for entire facts, is based on document-local and global information.

Proceedings of HLT/EMNLP 2005 Demonstration Abstracts, pages 22–23,
Vancouver, October 2005.

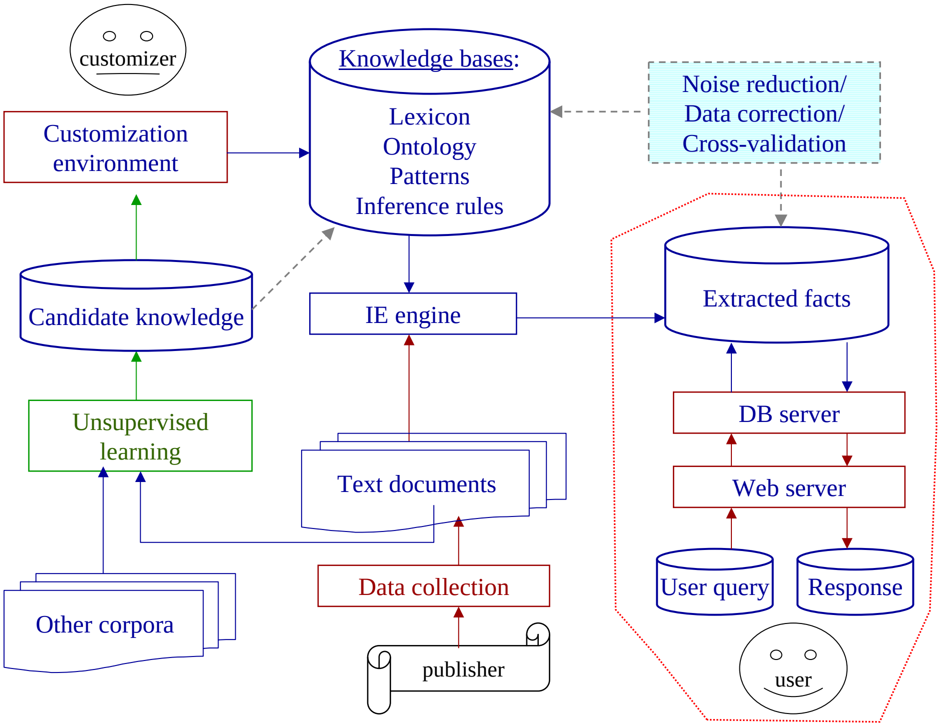


Figure 1: System architecture of ProMED-PLUS

acquisition, (Yangarber et al., 2002; Yangarber, 2003) requires a large corpus of domain-specific and general-topic texts. On the other hand, automatic error reduction requires a critical mass of extracted facts. Tighter integration between IE and KDD components, for mutual benefit, is advocated in recent related research, e.g., (Nahm and Mooney, 2000; McCallum and Jensen, 2003). In this system we have demonstrated that redundancy in the extracted data (despite the noise) can be leveraged to improve quality, by analyzing global trends and correcting erroneous fills which are due to local mis-analysis, (Yangarber and Jokipii, 2005). For this kind of approach to work, it is necessary to aggregate over a large body of extracted records.

The interface to the DB is accessible on-line at `doremi.cs.helsinki.fi/plus/` (lower-right of Fig. 1). It allows the user to view, select and sort the extracted outbreaks, as well as the individual incidents that make up the aggregated outbreaks. All facts in the database are linked back to the original reports from which they were extracted. The distribution of the outbreaks may also be plotted and queried through the *Geographic Map* view.

References

R. Grishman, S. Huttunen, and R. Yangarber. 2003. Information extraction for enhanced access to disease outbreak reports. *J. of Biomed. Informatics*, **35**(4).

A. McCallum and D. Jensen. 2003. A note on the unification of information extraction and data mining using conditional-probability, relational models. In *IJCAI'03 Workshop on Learning Statistical Models from Relational Data*.

U. Y. Nahm and R. Mooney. 2000. A mutually beneficial integration of data mining and information extraction. In *AAAI-2000*, Austin, TX.

R. Yangarber and L. Jokipii. 2005. Redundancy-based correction of automatically extracted facts. In *Proc. HLT-EMNLP 2005*, Vancouver, Canada.

R. Yangarber, W. Lin, and R. Grishman. 2002. Unsupervised learning of generalized names. In *Proc. COLING-2002*, Taipei, Taiwan.

R. Yangarber. 2003. Counter-training in discovery of semantic patterns. In *Proc. ACL-2003*, Sapporo, Japan.

A Flexible Conversational Dialog System for MP3 Player

Fuliang Weng[1] Lawrence Cavedon[2] Badri Raghunathan[1] Danilo Mirkovic[2] Ben Bei[1]
Heather Pon-Barry[1] Harry Bratt[3] Hua Cheng[2] Hauke Schmidt[1] Rohit Mishra[4] Brian Lathrop[4]
Qi Zhang[1] Tobias Scheideck[1] Kui Xu[1] Tess Hand-Bender[1] Sandra Upson[1] Stanley Peters[2]
Liz Shriberg[3] Carsten Bergmann[4]

Research and Technology Center, Robert Bosch Corp., Palo Alto, California[1]
Center for Study of Language and Information, Stanford University, Stanford, California[2]
Speech Technology and Research Lab, SRI International, Menlo Park, California[3]
Electronics Research Lab, Volkswagen of America, Palo Alto, California[4]

{Fuliang.weng,badri.raghunathan,hauke.Schmidt}@rtc.bosch.com
{lcavedon,huac,peters}@csli.Stanford.edu
{harry,ees}@speech.sri.com
{rohit.mishra,carsten.bergmann}@vw.com

1 Abstract

In recent years, an increasing number of new devices have found their way into the cars we drive. Speech-operated devices in particular provide a great service to drivers by minimizing distraction, so that they can keep their hands on the wheel and their eyes on the road. This presentation will demonstrate our latest development of an in-car dialog system for an MP3 player designed under a joint research effort from Bosch RTC, VW ERL, Stanford CSLI, and SRI STAR Lab funded by NIST ATP [Weng et al 2004] with this goal in mind. This project has developed a number of new technologies, some of which are already incorporated in the system. These include: end-pointing with prosodic cues, error identification and recovering strategies, flexible multi-threaded, multi-device dialog management, and content optimization and organization strategies. A number of important language phenomena are also covered in the system's functionality. For instance, one may use words relying on context, such as 'this,' 'that,' 'it,' and 'them,' to reference items mentioned in particular use contexts. Different types of verbal revision are also permitted by the system, providing a great convenience to its users. The system supports multi-threaded dialogs so that users can diverge to a different topic before the current one is finished and still come back to the first after the second topic is done. To lower the cognitive load on the drivers, the content optimization component organizes any information given to users based on ontological structures, and may also refine users' queries via various strategies. Domain knowledge is represented using OWL, a web ontology language recommended by W3C, which should greatly facilitate its portability to new domains.

The spoken dialog system consists of a number of components (see Fig. 1 for details). Instead of the hub architecture employed by Communicator projects [Senef et al, 1998], it is developed in Java and uses a flexible event-based, message-oriented middleware. This allows for dynamic registration of new components. Among the component modules in Figure 1, we use the Nuance speech recognition engine with class-based ngrams and dynamic grammars, and the Nuance Vocalizer as the TTS engine. The Speech Enhancer removes noises and echo. The Prosody module will provide additional features to the Natural Language Understanding (NLU) and Dialogue Manager (DM) modules to improve their performance.

The NLU module takes a sequence of recognized words and tags, performs a deep linguistic analysis with probabilistic models, and produces an XML-based semantic feature structure representation. Parallel to the deep analysis, a topic classifier assigns top n topics to the utterance, which are used in the cases where the dialog manager cannot make

Proceedings of HLT/EMNLP 2005 Demonstration Abstracts, pages 24–25,
Vancouver, October 2005.

any sense of the parsed structure. The NLU module also supports dynamic updates of the knowledge base.

The CSLI DM module mediates and manages interaction. It uses the dialogue-move approach to maintain dialogue context, which is then used to interpret incoming utterances (including fragments and revisions), resolve NPs, construct salient responses, track issues, etc. Dialogue states can also be used to bias SR expectation and improve SR performance, as has been performed in previous applications of the DM. Detailed descriptions of the DM can be found in [Lemon et al 2002; Mirkovic & Cavedon 2005].

The Knowledge Manager (KM) controls access to knowledge base sources (such as domain knowledge and device information) and their updates. Domain knowledge is structured according to domain-dependent ontologies. The current KM makes use of OWL, a W3C standard, to represent the ontological relationships between domain entities. Protégé (http://protege.stanford.edu), a domain-independent ontology tool, is used to maintain the ontology offline. In a typical interaction, the DM converts a user's query into a semantic frame (i.e. a set of semantic constraints) and sends this to the KM via the content optimizer.

The Content Optimization module acts as an intermediary between the dialogue management module and the knowledge management module during the query process. It receives semantic frames from the DM, resolves possible ambiguities, and queries the KM. Depending on the items in the query result as well as the configurable properties, the module selects and performs an appropriate optimization strategy.

Early evaluation shows that the system has a task completion rate of 80% on 11 tasks of MP3 player domain, ranging from playing requests to music database queries. Porting to a restaurant selection domain is currently under way.

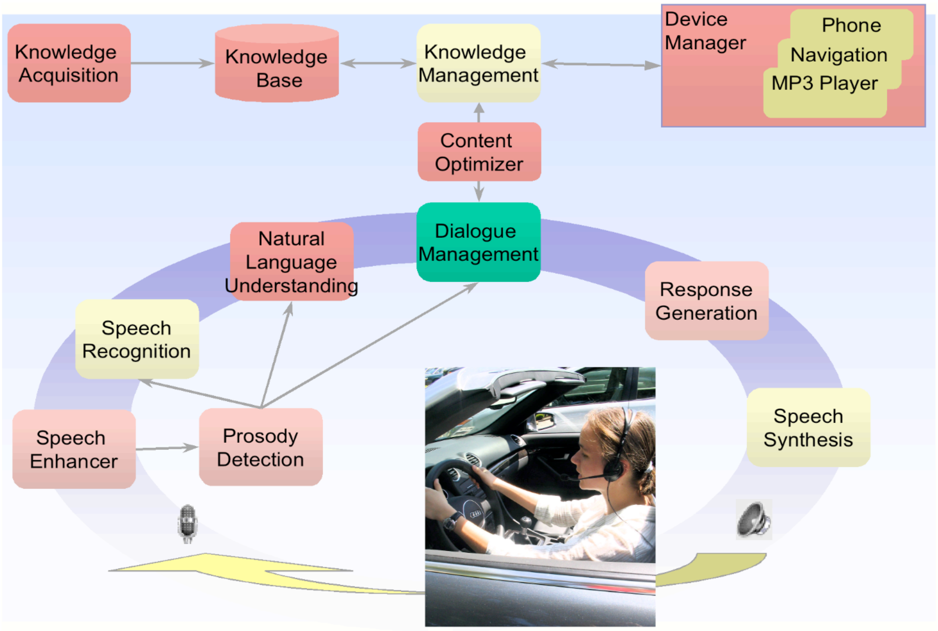
References

Seneff, Stephanie, Ed Hurley, Raymond Lau, Christine Pao, Philipp Schmid, and Victor Zue, *GALAXY-II: A Reference Architecture for Conversational System Development,* International Conference on Spoken Language Processing (ICSLP), Sydney, Australia, December 1998.

Lemon, Oliver, Alex Gruenstein, and Stanley Peters, *Collaborative activities and multi-tasking in dialogue systems*, Traitement Automatique des Langues (TAL), 43(2), 2002.

Mirkovic, Danilo, and Lawrence Cavedon, *Practical Multi-Domain, Multi-Device Dialogue Management*, Submitted for publication, April 2005.

Weng, Fuliang, Lawrence Cavedon, Badri Raghunathan, Hua Cheng, Hauke Schmidt, Danilo Mirkovic, et al., *Developing a conversational dialogue system for cognitively overloaded users,* International Conference on Spoken Language Processing (ICSLP), Jeju, Korea, October 2004.



Japanese Speech Understanding Using Grammar Specialization

Manny Rayner, Nikos Chatzichrisafis, Pierrette Bouillon
University of Geneva, TIM/ISSCO
40 bvd du Pont-d'Arve, CH-1211 Geneva 4, Switzerland
mrayner@riacs.edu
{Pierrette.Bouillon,Nikolaos.Chatzichrisafis}@issco.unige.ch

Yukie Nakao, Hitoshi Isahara, Kyoko Kanzaki
National Institute of Information and Communications Technology
3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto, Japan 619-0289
yukie-n@khn.nict.go.jp, {isahara,kanzaki}@nict.go.jp

Beth Ann Hockey
UCSC/NASA Ames Research Center
Moffet Field, CA 94035
bahockey@riacs.edu

Marianne Santaholma, Marianne Starlander
University of Geneva, TIM/ISSCO
40 bvd du Pont-d'Arve
CH-1211 Geneva 4, Switzerland
Marianne.Santaholma@eti.unige.ch
Marianne.Starlander@eti.unige.ch

The most common speech understanding architecture for spoken dialogue systems is a combination of speech recognition based on a class N-gram language model, and robust parsing. For many types of applications, however, grammar-based recognition can offer concrete advantages. Training a good class N-gram language model requires substantial quantities of corpus data, which is generally not available at the start of a new project. Head-to-head comparisons of class N-gram/robust and grammar-based systems also suggest that users who are familiar with system coverage get better results from grammar-based architectures (Knight et al., 2001). As a consequence, deployed spoken dialogue systems for real-world applications frequently use grammar-based methods. This is particularly the case for speech translation systems. Although leading research systems like Verbmobil and NESPOLE! (Wahlster, 2000; Lavie et al., 2001) usually employ complex architectures combining statistical and rule-based methods, successful practical examples like Phraselator and S-MINDS (Phraselator, 2005; Sehda, 2005) are typically phrasal translators with grammar-based recognizers.

Voice recognition platforms like the Nuance Toolkit provide CFG-based languages for writing grammar-based language models (GLMs), but it is challenging to develop and maintain grammars consisting of large sets of ad hoc phrase-structure rules. For this reason, there has been considerable interest in developing systems that permit language models be specified in higher-level formalisms, normally some kind of unification grammar (UG), and then compile these grammars down to the low-level platform formalisms. A prominent early example of this approach is the Gemini system (Moore, 1998).

Gemini raises the level of abstraction significantly, but still assumes that the grammars will be domain-dependent. In the Open Source REGULUS project (Regulus, 2005; Rayner et al., 2003), we have taken a further step in the direction of increased abstraction, and derive all recognizers from a single linguistically motivated UG. This derivation procedure starts with a large, application-independent UG for a language. An application-specific UG is then derived using an Explanation Based Learning (EBL) specialization technique. This corpus-based specialization process is parameterized by the training corpus and operationality criteria. The training corpus, which can be relatively small, consists of examples of utterances that should be recognized by the target application. The sentences of the corpus are parsed using the general grammar, then those parses are partitioned into phrases based on the operationality criteria. Each phrase defined by the operationality criteria is flattened, producing rules of a phrasal grammar for the application domain. This application-specific UG is then compiled into

Proceedings of HLT/EMNLP 2005 Demonstration Abstracts, pages 26–27,
Vancouver, October 2005.

a CFG, formatted to be compatible with the Nuance recognition platform. The CFG is compiled into the runtime recognizer using Nuance tools.

Previously, the REGULUS grammar specialization programme has only been implemented for English. In this demo, we will show how we can apply the same methodology to Japanese. Japanese is structurally a very different language from English, so it is by no means obvious that methods which work for English will be applicable in this new context: in fact, they appear to work very well. We will demo the grammars and resulting recognizers in the context of Japanese → English and Japanese → French versions of the Open Source MedSLT medical speech translation system (Bouillon et al., 2005; MedSLT, 2005).

The generic problem to be solved when building any sort of recognition grammar is that syntax alone is insufficiently constraining; many of the real constraints in a given domain and use situation tend to be semantic and pragmatic in nature. The challenge is thus to include enough non-syntactic constraints in the grammar to create a language model that can support reliable domain-specific speech recognition: we sketch our solution for Japanese.

The basic structure of our current general Japanese grammar is as follows. There are four main groups of rules, covering NP, PP, VP and CLAUSE structure respectively. The NP and PP rules each assign a sortal type to the head constituent, based on the domain-specific sortal constraints defined in the lexicon. VP rules define the complement structure of each syntactic class of verb, again making use of the sortal features. There are also rules that allow a VP to combine with optional adjuncts, and rules which allow null constituents, in particular null subjects and objects. Finally, clause-level rules form a clause out of a VP, an optional subject and optional adjuncts. The sortal features constrain the subject and the complements combining with a verb, but the lack of constraints on null constituents and optional adjuncts still means that the grammar is very loose. The grammar specialization mechanism flattens the grammar into a set of much simpler structures, eliminating the VP level and only permitting specific patterns of null constituents and adjuncts licenced by the training corpus.

We will demo several different versions of the Japanese-input medical speech translation system, differing with respect to the target language and the recognition architecture used. In particular, we will show a) that versions based on the specialized Japanese grammar offer fast and accurate recognition on utterances within the intended coverage of the system (Word Error Rate around 5%, speed under 0.1×RT), b) that versions based on the original general Japanese grammar are much less accurate and more than an order of magnitude slower.

References

P. Bouillon, M. Rayner, N. Chatzichrisafis, B.A. Hockey, M. Santaholma, M. Starlander, Y. Nakao, K. Kanzaki, and H. Isahara. 2005. A generic multi-lingual open source platform for limited-domain medical speech translation. In *In Proceedings of the 10th Conference of the European Association for Machine Translation (EAMT)*, Budapest, Hungary.

S. Knight, G. Gorrell, M. Rayner, D. Milward, R. Koeling, and I. Lewin. 2001. Comparing grammar-based and robust approaches to speech understanding: a case study. In *Proceedings of Eurospeech 2001*, pages 1779–1782, Aalborg, Denmark.

A. Lavie, C. Langley, A. Waibel, F. Pianesi, G. Lazzari, P. Coletti, L. Taddei, and F. Balducci. 2001. Architecture and design considerations in NESPOLE!: a speech translation system for e-commerce applications. In *Proceedings of HLT: Human Language Technology Conference*, San Diego, California.

MedSLT, 2005. http://sourceforge.net/projects/medslt/. As of 9 June 2005.

R. Moore. 1998. Using natural language knowledge sources in speech recognition. In *Proceedings of the NATO Advanced Studies Institute*.

Phraselator, 2005. http://www.phraselator.com/. As of 9 June 2005.

M. Rayner, B.A. Hockey, and J. Dowding. 2003. An open source environment for compiling typed unification grammars into speech recognisers. In *Proceedings of the 10th EACL (demo track)*, Budapest, Hungary.

Regulus, 2005. http://sourceforge.net/projects/regulus/. As of 9 June 2005.

Sehda, 2005. http://www.sehda.com/. As of 9 June 2005.

W. Wahlster, editor. 2000. *Verbmobil: Foundations of Speech-to-Speech Translation*. Springer.

THE MIT SPOKEN LECTURE PROCESSING PROJECT

James R. Glass, Timothy J. Hazen, D. Scott Cyphers, Ken Schutte and Alex Park
The MIT Computer Science and Artificial Intelligence Laboratory
32 Vassar Street, Cambridge, Massachusetts, 02476, USA
{hazen,jrg,cyphers}@csail.mit.edu

Abstract

We will demonstrate the MIT Spoken Lecture Processing Server and an accompanying lecture browser that students can use to quickly locate and browse lecture segments that apply to their query. We will show how lecturers can upload recorded lectures and companion text material to our server for automatic processing. The server automatically generates a time-aligned word transcript of the lecture which can be downloaded for use within a browser. We will also demonstrate a browser we have created which allows students to quickly locate and browse audio segments that are relevant to their query. These tools can provide students with easier access to audio (or audio/visual) lectures, hopefully improving their educational experience.

1 Introduction

Over the past decade there has been increasing amounts of educational material being made available on-line. Projects such as MIT OpenCourseWare provide continuous worldwide access to educational materials to help satisfy our collective thirst for knowledge. While the majority of such material is currently text-based, we are beginning to see dramatic increases in the amount of audio and visual recordings of lecture material. Unlike text materials, untranscribed audio data can be tedious to browse, making it difficult to utilize the information fully without time-consuming data preparation. Moreover, unlike some other forms of spoken communication such as telephone conversations or television and radio broadcasts, lecture processing has until recently received little attention or benefit from the development of human language technology. The single biggest effort, to date, is on-going work in Japan using the Corpus of Spontaneous Japanese [1,3,4].

Lectures are particularly challenging for automatic speech recognizers because the vocabulary used within a lecture can be very technical and specialized, yet the speaking style can be very spontaneous. As a result, even if parallel text materials are available in the form of textbooks or related papers, there are significant linguistic differences between written and oral communication styles. Thus, it is a challenge to predict how a written passage might be spoken, and vice versa. By helping to focus a research spotlight on spoken lecture material, we hope to begin to overcome these and many other fundamental issues.

While audio-visual lecture processing will perhaps be ultimately most useful, we have initially focused our attention on the problem of spoken lecture processing. Within this realm there are many challenging research issues pertaining to the development of effective automatic transcription, indexing, and summarization. For this project, our goals have been to a) help create a corpus of spoken lecture material for the research community, b) analyze this corpus to better understand the linguistic characteristics of spoken lectures, c) perform speech recognition and information retrieval experiments on these data to benchmark performance on these data, d) develop a prototype spoken lecture processing server that will allow educators to automatically annotate their recorded lecture data, and e) develop prototype software that will allow students to browse the resulting annotated lectures.

2 Project Details

As mentioned earlier, we have developed a web-based Spoken Lecture Processing Server (http://groups.csail.mit.edu/sls/lectures) in which users can upload audio files for automatic transcription and indexing. In our work, we have ex-

Proceedings of HLT/EMNLP 2005 Demonstration Abstracts, pages 28–29,
Vancouver, October 2005.

perimented with collecting audio data using a small personal digital audio recorder (an iRiver N10). To help the speech recognizer, users can provide their own supplemental text files, such as journal articles, book chapters, etc., which can be used to adapt the language model and vocabulary of the system. Currently, the key steps of the transcription process are as follows: a) adapt a topic-independent vocabulary and language model using any supplemental text materials, b) automatically segment the audio file into short chunks of pause-delineated speech, and c) automatically annotate these chunks using a speech recognition system.

Language model adaptation is performed is two steps. First the vocabulary of any supplemental text material is extracted and added to an existing topic-independent vocabulary of nearly 17K words. Next, the recognizer merges topic-independent word sequence statistics from an existing corpus of lecture material with the topic-dependent statistics of the supplemental material to create a topic-adapted language model.

The segmentation algorithm is performed in two steps. First the audio file is arbitrarily broken into 10-second chunks for speech detection processing using an efficient speaker-independent phonetic recognizer. To help improve its speech detection accuracy, this recognizer contains models for non-lexical artifacts such as laughs and coughs as well as a variety of other noises. Contiguous regions of speech are identified from the phonetic recognition output (typically 6 to 8 second segments of speech) and passed alone to our speech recognizer for automatic transcription. The speech segmentation and transcription steps are currently performed in a distributed fashion over a bank of computation servers. Once recognition is completed, the audio data is indexed (based on the recognition output) in preparation for browsing by the user.

The lecture browser provides a graphical user interface to one or more automatically transcribed lectures. A user can type a text query to the browser and receive a list of hits within the indexed lectures. When a hit is selected, it is shown in the context of the lecture transcription. The user can adjust the duration of context preceding and following the hit, navigate to and from the preceding and following parts of the lecture, and listen to the displayed segment. Orthographic segments are highlighted as they are played.

3 Experimental Results

To date we have collected and analyzed a corpus of approximately 300 hours of audio lectures including 6 full MIT courses and 80 hours of seminars from the MIT World web site [2]. We are currently in the process of expanding this corpus. From manual transcriptions we have generated and verified time-aligned transcriptions for 169 hours of our corpus, and we are in the process of time-aligning transcriptions for the remainder of our corpus.

We have performed initial speech recognition experiments using 10 computer science lectures. In these experiments we have discovered that, despite high word error rates (in the area of 40%), retrieval of short audio segments containing important keywords and phrases can be performed with a high-degree of reliability (over 90% F-measure when examining precision and recall results) [5]. These results are similar in nature to the findings in the SpeechBot project (which performs a similar service for online broadcast news archives) [6].

References

[1] S. Furui, “Recent advances in spontaneous speech recognition and understanding,” in *Proc. ISCA & IEEE Workshop on Spontaneous Speech Processing and Recognition (SSPR)*, pp. 1-6, Tokyo, April 2003.

[2] J. Glass, T. Hazen, L. Hetherington, and C. Wang, “Analysis and Processing of Lecture Audio Data: Preliminary Investigations,” in *Proc. HLT/NAACL Speech Indexing Workshop*, 9-12, Boston, May 2004.

[3] T. Kawahara, H. Nanjo. And S. Furui, “Automatic transcription of spontaneous lecture speech,” in *IEEE Workshop on Automatic Speech Recognition and Understanding*, pp. 186-189, Trento, Italy, December 2001.

[4] H. Nanjo and T. Kawahara, “Language model and speaking rate adaptation for spontaneous speech recognition,” *IEEE Transactions of Speech and Audio Processing*, vol. 12, no. 4, pp. 391-400, July 2004.

[5] A. Park, T. Hazen, and J. Glass, "Automatic Processing of Audio Lectures for Information Retrieval: Vocabulary Selection and Language Modeling," Proc. ICASSP, Philadelphia, PA, March 2005.

[6] J.-M. Van Thong, *et al*, “SpeechBot: An experimental speech-based search engine for multimedia content on the web. *IEEE Transactions of Multimedia*, vol. 4, no. 1, pp. 88-96, March 2002.

MBOI: Discovery of Business Opportunities on the Internet
Extended Abstract

Arman Tajarobi, Jean-François Garneau
Nstein Technologies
Québec, Canada
{arman.tajarobi,jf.garneau}
@nstein.com

François Paradis
Université de Montréal
Québec, Canada
paradifr@iro.umontreal.ca

We propose a tool for the discovery of business opportunities on the Web, more specifically to help a user find relevant *call for tenders* (CFT), i.e. invitations to contractors to submit a tender for their products/services. Simple keyword-based Information Retrieval do not capture the relationships in the data, which are needed to answer the complex needs of the users. We therefore augment keywords with information extracted through natural language processing and business intelligence tools. As opposed to most systems, this information is used at all stages in the back-end and interface. The benefits are twofold: first we obtain higher precision of search and classification, and second the user gains access to a deeper level of information.

Two challenges are: how to discover new CFT and related documents on the Web, and how to extract information from these documents, knowing that the Web offers no guarantee on the structure and stability of those documents. A major hurdle to the discovery of new documents is the poor degree of "linkedness" between businesses, and the open topic area, which makes topic-focused Web crawling (Aggarwal et al., 2001) unapplicable. To extract information, *wrappers* (Soderland, 1999), i.e. tools that can recognise textual and/or structural patterns, have limited success because of the diversity and volatility of Web documents.

Since we cannot assume a structure for documents, we exploit information usually contained in CFTs: contracting authority, opening/closing date, location, legal notices, conditions of submission, classification, etc. These can appear marked up with tags or as free-text.

A first type of information to extract are the so-called *named entities* (Maynard et al., 2001), i.e. names of people, organisations, locations, time or quantities. To these standard entities we add some application-specific entities such as FAR (regulation number), product dimensions, etc. To extract named entities we use Nstein NFinder$^{\text{TM}}$, which uses a combination of lexical rules and a dictionary. More details about the entities, statistics and results can be found in (Paradis and Nie, 2005a).

We use another tool, Nstein Nconcept$^{\text{TM}}$, to extract *concepts*, which capture the "themes" or "relevant phrases" in a document. NConcept uses a combination of statistics and linguistic rules.

As mentioned above, CFTs not only contains information about the subject of the tender, but also procedural and regulation information. We tag passages in the document as "subject" or "non-subject", according to the presence or absence of the most discriminant bigrams. Some heuristics are also applied to use the "good predictors" such as URL and money, or to further refine the non-subject passages into "regulation". More details can be found in (Paradis and Nie, 2005b).

Another information to extract is the industry or service, according to a classification schema such as NAICS (North American Industry Classification System) or CPV (Common Procurement Vocabulary). We perform multi-schema, multi-label classification, which facilitates use across economic zones (for instance, an American user may not be familiar with CPV, a European standard) and confusion over schemas versions (NAICS version 1997/Canada vs. NAICS version 2002). Our classifier is a simple Naive Bayes, trained over 20,000 documents gathered from an American Government tendering site, FBO (Federal Business Opportunities). Since we have found classification to be sensitive to the pres-

Proceedings of HLT/EMNLP 2005 Demonstration Abstracts, pages 30–31,
Vancouver, October 2005.

ence of procedural contents, we remove non-subject passages, as tagged above. The resulting performance is 61% micro-F1 (Paradis and Nie, 2005b).

Finally, a second level of extraction is performed to infer information about organisations: their contacts, business relationships, spheres of activities, average size of contract, etc. This is refered to as *business intelligence* (Betts, 2003). For this extraction we not only use CFTs, but also awards (i.e. past information about successful bids) and news (i.e. articles published about an organisation). For news, we collect co-occurences of entities and classify them using a semantic network. For example, the passage “Sun vs. Microsoft” is evidence towards the two companies being competitors.

The extracted information is indexed and queried using *Apache Lucene.*, with a Web front-end served by *Jakarta Turbine*. The interface was designed to help the user make the most of the extracted information, whether in query formulation, document perusing, or navigation.

Our system supports precise queries by indexing free-text and extracted information separately. For example, the simple keyword query “bush” returns all documents where the word occurs, including documents about bush trimming and president Bush, while the query “person:Bush” only returns documents about President Bush. However such queries are not very user-friendly. We thus provide an interface for advanced queries and query refinement.

The extracted information from the 100 top query results is gathered and presented in small scrollable lists, one for each entity type. For example, starting with keyword “bush”, the user sees a list of people in the “person” box, and could choose “Bush” to refine her query. The list is also used to expand the query with a related concept (for example, “removal services” is suggested for “snow”), the expansion of an acronym, etc.

Queries can be automatically translated using Cross-Language Information Retrieval techniques (Peters et al., 2003). To this end we have built a statistical translation model trained from a collection of 100,000 French-English pair documents from a European tendering site, TED (Tenders Electronic Daily). Two dictionaries were built: one with simple terms, and one with “concepts”, extracted as above. The intuition is that simple terms will offer better recall while concepts will give better precision.

The interface shows and allows navigation to the extracted information. When viewing a CFT, the user can highlight the entities, as well as the subject and regulation passages. She can also click on an organisation to get a company profile, which shows the business intelligence attributes as well as related documents such as past awards or news.

We are currently expanding the business intelligence functionalities, and implementing user “profiles”, which will save contextual or background information and use it transparently to affect querying.

Acknowledgments

This project was financed jointly by Nstein Technologies and NSERC.

References

Charu C. Aggarwal, Fatima Al-Garawi, and Philip S. Yu. 2001. Intelligent crawling on the world wide web with arbitrary predicates. In *Proceedings International WWW Conference*.

Mitch Betts. 2003. The future of business intelligence. *Computer World*, 14 April.

D. Maynard, V. Tablan, C. Ursu, H. Cunningham, and Y. Wilks. 2001. Named entity recognition from diverse text types. In *Recent Advances in Natural Language Processing*, pages 257–274.

François Paradis and Jian-Yun Nie. 2005a. Discovery of business opportunities on the internet with information extraction. In *IJCAI-05 Workshop on Multi-Agent Information Retrieval and Recommender Systems*, 31 July.

François Paradis and Jian-Yun Nie. 2005b. Filtering contents with bigrams and named entities to improve text classification. In *Asia Information Retrieval Symposium*, 13–15 October.

C. Peters, M. Braschler, J. Gonzalo, and M. Kluck, editors. 2003. *Advances in Cross-Language Information Retrieval Systems*. Springer.

Stephen Soderland. 1999. Learning information extraction rules for semi-structured and free text. *Machine Learning*, 44(1).

OPINE: Extracting Product Features and Opinions from Reviews

Ana-Maria Popescu **Bao Nguyen** **Oren Etzioni**
Department of Computer Science and Engineering
University of Washington
Seattle, WA 98195-2350
{amp,omicron,etzioni}@cs.washington.edu

Abstract

Consumers have to often wade through a large number of on-line reviews in order to make an informed product choice. We introduce OPINE, an unsupervised, high-precision information extraction system which mines product reviews in order to build a model of product features and their evaluation by reviewers.

```
Input: product class C, reviews R.
Output: set of [feature, ranked opinion list] tuples
R' ← parseReviews(R);
E ← findExplicitFeatures(R', C);
O ← findOpinions(R', E);
CO ← clusterOpinions(O);
I ← findImplicitFeatures(CO, E);
RO ← rankOpinions(CO);
$\{(f, o_i, ...o_j)\}$←outputTuples(RO, I∪E);
```

Figure 1: OPINE **Overview.**

1 Introduction

The Web contains a wealth of customer reviews - as a result, the problem of "review mining" has seen increasing attention over the last few years from (Turney, 2003; Hu and Liu, 2004) and many others. We decompose the problem of review mining into the following subtasks: **a) Identify product features, b) Identify opinions regarding product features, c) Determine the polarity of each opinion** and **d) Rank opinions according to their strength** (e.g., "abominable" is stronger than "bad").

We introduce OPINE, an unsupervised information extraction system that embodies a solution to each of the above subtasks. The remainder of this paper is organized as follows: Section 2 describes OPINE's components together with their experimental evaluation and Section 3 describes the related work.

2 OPINE Overview

OPINE is built on top of KNOWITALL, a Web-based, domain-independent information extraction system (Etzioni et al., 2005). Given a set of relations of interest, KNOWITALL instantiates relation-specific generic extraction patterns into extraction rules which find candidate facts. The Assessor module then assigns a probability to each candidate using a form of *Point-wise Mutual Information* (PMI) between phrases that is estimated from Web search engine hit counts (Turney, 2003). It computes the PMI between each fact and *discriminator phrases* (e.g., "is a scanner" for the `isA()` relationship in the context of the `Scanner` class). Given fact f and discriminator d, the computed PMI score is:

$$\text{PMI}(f, d) = \frac{\text{Hits}(d + f)}{\text{Hits}(d)*\text{Hits}(f)}$$

The PMI scores are converted to binary features for a Naive Bayes Classifier, which outputs a probability associated with each fact.

Given product class C with instances I and reviews R, OPINE's goal is to find the set of (feature, opinions) tuples $\{(f, o_i, ...o_j)\}$ s.t. $f \in F$ and $o_i, ...o_j \in O$, where:

a) F is the set of product class features in R.

b) O is the set of opinion phrases in R.

c) opinions associated with a particular feature are ranked based on their strength.

OPINE's solution to this task is outlined in Figure 1. In the following, we describe in detail each step.

Explicit Feature Extraction OPINE parses the reviews using the MINIPAR dependency parser (Lin, 1998) and applies a simple pronoun-resolution module to the parsed data. The system then finds *explicitly mentioned* product features (E) using an extended version of KNOWITALL's extract-and-assess strategy described above. OPINE extracts the following types of *product features*: properties, parts, features of product parts (e.g., `ScannerCoverSize`), *related concepts* (e.g., `Image`

Proceedings of HLT/EMNLP 2005 Demonstration Abstracts, pages 32–33,
Vancouver, October 2005.

is related to Scanner) and parts and properties of related concepts (e.g., ImageSize). When compared on this task with the most relevant previous review-mining system in (Hu and Liu, 2004), OPINE obtains a 22% improvement in precision with only a 3% reduction in recall on the relevant 5 datasets. One third of this increase is due to OPINE's feature assessment step and the rest is due to the use of Web PMI statistics.

Opinion Phrases OPINE extracts adjective, noun, verb and adverb phrases attached to explicit features as potential opinion phrases. OPINE then collectively assigns *positive*, *negative* or *neutral* semantic orientation (SO) labels to their respective head words. This problem is similar to labeling problems in computer vision and OPINE uses a well-known computer vision technique, *relaxation labeling*, as the basis of a 3-step SO label assignment procedure. First, OPINE identifies the average SO label for a word w *in the context of the review set*. Second, OPINE identifies the average SO label for each word w *in the context of a feature* f *and of the review set* ("hot" has a negative connotation in "hot room", but a positive one in "hot water"). Finally, OPINE identifies the SO label of word w *in the context of feature* f *and sentence* s. For example, some people like large scanners ("I love this large scanner") and some do not ("I hate this large scanner"). The phrases with non-neutral head words are retained as opinion phrases and their polarity is established accordingly. On the task of opinion phrase extraction, OPINE obtains a precision of 79% and a recall of 76% and on the task of opinion phrase polarity extraction OPINE obtains a precision of 86% and a recall of 84%.

Implicit Features Opinion phrases refer to *properties*, which are sometimes *implicit* (e.g., "tiny phone" refers to the phone size). In order to extract such properties, OPINE first clusters opinion phrases (e.g., *tiny* and *small* will be placed in the same cluster), automatically labels the clusters with property names (e.g., *Size*) and uses them to build *implicit* features (e.g., PhoneSize). Opinion phrases are clustered using a mixture of WordNet information (e.g., antonyms are placed in the same cluster) and lexical pattern information (e.g., "clean, *almost* spotless" suggests that "clean" and "spotless" are likely to refer to the same property). (Hu and Liu, 2004) doesn't handle implicit features, so we have evaluated the impact of implicit feature extraction on two separate sets of reviews in the Hotels and Scanners domains. Extracting implicit features (in addition to explicit features) has resulted in a 2% increase in precision and a 6% increase in recall for OPINE on the task of feature extraction.

Ranking Opinion Phrases Given an opinion cluster, OPINE uses the final probabilities associated with the SO labels in order to derive an initial opinion phrase strength ranking (e.g., $great > good > average$) in the manner of (Turney, 2003). OPINE then uses Web-derived constraints on the relative strength of phrases in order to improve this ranking. Patterns such as "a_1, (*) even a_2" are good indicators of how strong a_1 is relative to a_2. OPINE bootstraps a set of such patterns and instantiates them with pairs of opinions in order to derive constraints such as $strength(deafening) > strength(loud)$. OPINE also uses synonymy and antonymy-based constraints such as $strength(clean) = strength(dirty)$. The constraint set induces a constraint satisfaction problem whose solution is a ranking of the respective cluster opinions (the remaining opinions maintain their default ranking). OPINE's accuracy on the opinion ranking task is 87%. Finally, OPINE outputs a set of (feature, ranked opinions) tuples for each product.

3 Related Work

The previous review-mining systems most relevant to our work are (Hu and Liu, 2004) and (Kobayashi et al., 2004). The former's precision on the explicit feature extraction task is 22% lower than OPINE's while the latter employs an iterative semi-automatic approach which requires significant human input; neither handles *implicit* features. Unlike previous research on identifying the subjective character and the polarity of phrases and sentences ((Hatzivassiloglou and Wiebe, 2000; Turney, 2003) and many others), OPINE identifies the context-sensitive polarity of opinion phrases. In contrast to supervised methods which distinguish among strength levels for sentences or clauses ((Wilson et al., 2004) and others), OPINEuses an unsupervised constraint-based opinion ranking approach.

References

O. Etzioni, M. Cafarella, D. Downey, S. Kok, A. Popescu, T. Shaked, S. Soderland, D. Weld, and A. Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence*, 165(1):91–134.

V. Hatzivassiloglou and J. Wiebe. 2000. Effects of Adjective Orientation and Gradability on Sentence Subjectivity. In *COLING*, pages 299–305.

M. Hu and B. Liu. 2004. Mining and Summarizing Customer Reviews. In *KDD*, pages 168–177, Seattle, WA.

N. Kobayashi, K. Inui, K. Tateishi, and T. Fukushima. 2004. Collecting Evaluative Expressions for Opinion Extraction. In *IJCNLP*, pages 596–605.

D. Lin. 1998. Dependency-based evaluation of MINIPAR. In *Workshop on Evaluation of Parsing Systems at ICLRE*.

P. Turney. 2003. Inference of Semantic Orientation from Association. In *CoRR cs. CL/0309034*.

T. Wilson, J. Wiebe, and R. Hwa. 2004. Just how mad are you? finding strong and weak opinion clauses. In *AAAI*, pages 761–769.

OpinionFinder: A system for subjectivity analysis

Theresa Wilson‡, Paul Hoffmann‡, Swapna Somasundaran†, Jason Kessler†, Janyce Wiebe†‡, Yejin Choi§, Claire Cardie§, Ellen Riloff*, Siddharth Patwardhan*

‡Intelligent Systems Program, University of Pittsburgh, Pittsburgh, PA 15260
†Department of Computer Science, University of Pittsburgh, Pittsburgh, PA 15260
§Department of Computer Science, Cornell University, Ithaca, NY 14853
*School of Computing, University of Utah, Salt Lake City, UT 84112

{twilson,hoffmanp,swapna,jsk44,wiebe}@cs.pitt.edu,
{ychoi,cardie}@cs.cornell.edu, {riloff,sidd}@cs.utah.edu

1 Introduction

OpinionFinder is a system that performs *subjectivity analysis*, automatically identifying when opinions, sentiments, speculations, and other *private states* are present in text. Specifically, OpinionFinder aims to identify *subjective* sentences and to mark various aspects of the subjectivity in these sentences, including the *source* (holder) of the subjectivity and words that are included in phrases expressing positive or negative sentiments.

Our goal with OpinionFinder is to develop a system capable of supporting other Natural Language Processing (NLP) applications by providing them with information about the subjectivity in documents. Of particular interest are question answering systems that focus on being able to answer opinion-oriented questions, such as the following:

> How is Bush's decision not to ratify the Kyoto Protocol looked upon by Japan and other US allies?
>
> How do the Chinese regard the human rights record of the United States?

To answer these types of questions, a system needs to be able to identify when opinions are expressed in text and who is expressing them. Other applications that would benefit from knowledge of subjective language include systems that summarize the various viewpoints in a document or that mine product reviews. Even typical fact-oriented applications, such as information extraction, can benefit from subjectivity analysis by filtering out opinionated sentences (Riloff et al., 2005).

2 OpinionFinder

OpinionFinder runs in two modes, batch and interactive. Document processing is largely the same for both modes. In batch mode, OpinionFinder takes a list of documents to process. Interactive mode provides a front-end that allows a user to query on-line news sources for documents to process.

2.1 System Architecture Overview

OpinionFinder operates as one large pipeline. Conceptually, the pipeline can be divided into two parts. The first part performs mostly general purpose document processing (e.g., tokenization and part-of-speech tagging). The second part performs the subjectivity analysis. The results of the subjectivity analysis are returned to the user in the form of SGML/XML markup of the original documents.

2.2 Document Processing

For general document processing, OpinionFinder first runs the Sundance partial parser (Riloff and Phillips, 2004) to provide semantic class tags, identify Named Entities, and match extraction patterns that correspond to subjective language (Riloff and Wiebe, 2003). Next, OpenNLP[1] 1.1.0 is used to tokenize, sentence split, and part-of-speech tag the data, and the Abney stemmer[2] is used to stem. In batch mode, OpinionFinder parses the data again, this time to obtain constituency parse trees (Collins, 1997), which are then converted to dependency parse trees (Xia and Palmer, 2001). Currently, this stage is only

[1] http://opennlp.sourceforge.net/

[2] SCOL version 1g available at http://www.vinartus.net/spa/

Proceedings of HLT/EMNLP 2005 Demonstration Abstracts, pages 34–35,
Vancouver, October 2005.

available for batch mode processing due to the time required for parsing. Finally, a clue-finder is run to identify words and phrases from a large subjective language lexicon.

2.3 Subjectivity Analysis

The subjectivity analysis has four components.

2.3.1 Subjective Sentence Classification

The first component is a Naive Bayes classifier that distinguishes between subjective and objective sentences using a variety of lexical and contextual features (Wiebe and Riloff, 2005; Riloff and Wiebe, 2003). The classifier is trained using subjective and objective sentences, which are automatically generated from a large corpus of unannotated data by two high-precision, rule-based classifiers.

2.3.2 Speech Events and Direct Subjective Expression Classification

The second component identifies speech events (e.g., “said,” “according to”) and direct subjective expressions (e.g., “fears,” “is happy”). Speech events include both speaking and writing events. Direct subjective expressions are words or phrases where an opinion, emotion, sentiment, etc. is directly described. A high-precision, rule-based classifier is used to identify these expressions.

2.3.3 Opinion Source Identification

The third component is a source identifier that combines a Conditional Random Field sequence tagging model (Lafferty et al., 2001) and extraction pattern learning (Riloff, 1996) to identify the sources of speech events and subjective expressions (Choi et al., 2005). The source of a speech event is the speaker; the source of a subjective expression is the experiencer of the private state. The source identifier is trained on the MPQA Opinion Corpus[3] using a variety of features. Because the source identifier relies on dependency parse information, it is currently only available in batch mode.

2.3.4 Sentiment Expression Classification

The final component uses two classifiers to identify words contained in phrases that express positive or negative sentiments (Wilson et al., 2005). The first classifier focuses on identifying sentiment expressions. The second classifier takes the sentiment expressions and identifies those that are positive and negative. Both classifiers were developed using BoosTexter (Schapire and Singer, 2000) and trained on the MPQA Corpus.

3 Related Work

Please see (Wiebe and Riloff, 2005; Choi et al., 2005; Wilson et al., 2005) for discussions of related work in automatic opinion and sentiment analysis.

4 Acknowledgments

This work was supported by the Advanced Research and Development Activity (ARDA), by the NSF under grants IIS-0208028, IIS-0208798 and IIS-0208985, and by the Xerox Foundation.

References

Y. Choi, C. Cardie, E. Riloff, and S. Patwardhan. 2005. Identifying sources of opinions with conditional random fields and extraction patterns. In *HLT/EMNLP 2005*.

M. Collins. 1997. Three generative, lexicalised models for statistical parsing. In *ACL-1997*.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML-2001*.

E. Riloff and W. Phillips. 2004. An Introduction to the Sundance and AutoSlog Systems. Technical Report UUCS-04-015, School of Computing, University of Utah.

E. Riloff and J. Wiebe. 2003. Learning extraction patterns for subjective expressions. In *EMNLP-2003*.

E. Riloff, J. Wiebe, and W. Phillips. 2005. Exploiting subjectivity classification to improve information extraction. In *AAAI-2005*.

E. Riloff. 1996. An Empirical Study of Automated Dictionary Construction for Information Extraction in Three Domains. *Artificial Intelligence*, 85:101–134.

R. E. Schapire and Y. Singer. 2000. BoosTexter: A boosting-based system for text categorization. *Machine Learning*, 39(2/3):135–168.

J. Wiebe and E. Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. In *CICLing-2005*.

T. Wilson, J. Wiebe, and P. Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *HLT/EMNLP 2005*.

F. Xia and M. Palmer. 2001. Converting dependency structures to phrase structures. In *HLT-2001*.

[3] The MPQA Opinion Corpus can be freely obtained at http://nrrc.mitre.org/NRRC/publications.htm.

POSBIOTM/W: A Development Workbench For Machine Learning Oriented Biomedical Text Mining System [*]

Kyungduk Kim, Yu Song, 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
{getta, songyu, gblee}@postech.ac.kr

Abstract

The POSBIOTM/W[1] is a workbench for machine-learning oriented biomedical text mining system. The POSTBIOTM/W is intended to assist biologist in mining useful information efficiently from biomedical text resources. To do so, it provides a suit of tools for gathering, managing, analyzing and annotating texts. The workbench is implemented in Java, which means that it is platform-independent.

1 Introduction

Large amounts of biomedical literature exist and the volume continues to grow exponentially. Following the increase of literature, there is growing need for appropriate tools in support of collecting, managing, creating, annotating and exploiting rich biomedical text resources.

Especially, information on interactions among biological entities is very important for understanding the biological process in a living cell (Blascheke et. al., 1999). In our POSBIOTM/W workbench, we use a supervised machine learning method to generate rules automatically to extract biological events from free texts with minimum human effort. And we adopt the Conditional Random Fields (CRF) model (Lafferty et. al.,2001) for the biomedical named-entity recognition (NER) task. Finally, to reduce the labeling effort in a larger extent we incorporate an active learning idea into the workbench.

2 System Description

The POSBIOTM/W comprises a set of appropriate tools to provide users a convenient environment for gathering, managing and analyzing biomedical text and for named-entity annotation. The workbench consists of four components: Managing tool, NER tool, Event Extraction Tool and Annotation Tool. And we adopt an active learning idea into the workbench to improve the NER and the Event Extraction module's performance. The overall design is shown in Figure 1.

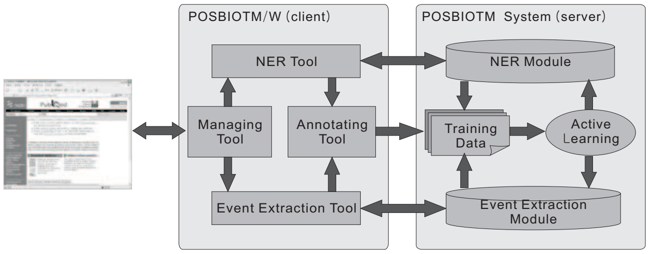


Figure 1: Overview of POSBIOTM/W

2.1 Managing tool

Main objective of the Managing tool is to help biologists search, collect and manage literatures relevant to their interest. Users can access to the PubMed database of bibliographic information using quick searching bar and incremental PubMed search engine.

[*] The research was supported by Brain Neuro Informatics Research program by MOCIE.

[1] POSBIOTM/W stands for POSTECH Bio-Text Mining System Workbench

Proceedings of HLT/EMNLP 2005 Demonstration Abstracts, pages 36–37,
Vancouver, October 2005.

2.2 NER tool

The NER tool is a client tool of POSBIOTM-NER module and able to automatically annotate biomedical-related texts. The NER tool provides access to three target-specific named entity models - GENIA-NER model, GENE-NER model and GPCR-NER model. Each of these model is trained based on GENIA-Corpus (Kim et. al., 2003), BioCreative data (Blaschke et. al., 2004) and POSBIOTM/NE corpus[2] respectively. In POSBIOTM-NER system, we adopt the Conditional Random Fields (CRF) model (Lafferty et. al., 2001) for the biomedical NER task.

2.3 Event Extraction tool

The Event Extraction tool extracts several biological events from texts using automatically generated rules. We use a supervised machine learning method to overcome a knowledge-engineering bottleneck by learning event extraction rules automatically. We modify the WHISK (Soderland, 1999) algorithm to provide a two-level rule learning method as a divide-and-conquer strategy. In two-level rule learning, the system learns event extraction rules which are inside of the noun chunk at first level, and then it learns the rules for whole sentence.

Since the system extracts biological events using automatically generated rules, we can not guarantee that every extracted event is always correct because many different rules can be applied to the same sentence. Therefore we try to verify the result with a Maximum Entropy (ME) classifier to remove incorrectly extracted events. For each extracted event, we verify each component of the event with the ME classifier model. If one component is contradicted to the class assigned by the classification model, we will remove the event. For detail event extraction process, please consult our previous paper (Kim et. al., 2004).

2.4 Annotation tool

Our workbench provides a Graphical User Interface based Annotation tool which enables the users to annotate and correct the result of the named-entity recognition and the event extraction. And users can upload the revised data to the POSBIOTM system, which would contribute to the incremental build-up of named-entity and relation annotation corpus.

2.5 Active learning

To minimize the human labeling effort, we employ the active learning method to select the most informative samples. We proposed a new active learning paradigm which considers not only the uncertainty of the classifier but also the diversity of the corpus, which will soon be published.

References

Christian Blaschke, Andrade, M.A., Ouzouis, C., Valencia, A.. 1999. *Automatic extraction of biological information from scientific text : protein-protein interactions*. Intelligent Systems for Molecular Biology 60-67.

Christian Blaschke, L. Hirschman, and A. Yeh, editors. 2004. *Proceedings of the BioCreative Workshop, Granda, March.* `http://www.pdg.cnb.uam.es/BioLINK/workshop_BioCreative_04/handout/`

Eunju Kim, Yu Song, Gary Geunbae Lee, Byoung-Kee Yi. 2004. *Learning for interaction extraction and verification from biological full articles*. Proceedings of the ACM SIGIR 2004 workshop on search and discovery in bioinformatics, July 2004, Sheffield, UK

J.-D. Kim, T. Ohta, Y. Tateisi and J. Tsujii 2003. *GENIA corpus - a semantically annotated corpus for biotextmining*. Bioinformatics, Vol 19 Suppl. 1 2003, pages i180-i182

J. Lafferty, A. McCallum and F. Pereira 2001. *Conditional random fields: probabilistic models for segmenting and labelling sequence data*. International Conference on Machine Learning.

Soderland S. 1999. *Learning information extraction rules for semi-structured and free text*. Machine Learning, volume 34, 233-272.

[2] POSBIOTM/NE corpus, our own corpus, is used to identify four target named entities: protein, gene, small molecule and cellular process.

Author Index