Coling 2010

23rd International Conference on Computational Linguistics

Demonstrations Volume

23 August – 27 August 2010
Beijing International Convention Center
Beijing, China

Produced by
Chinese Information Processing Society of China

To order the CD of Coling 2010 and its Workshop Proceedings, please contact:

Chinese Information Processing Society of China
No.4, Southern Fourth Street
Haidian District, Beijing, 100190
China
Tel: +86-010-62562916
Fax: +86-010-62562916
cips@iscas.ac.cn

Table of Contents

A Paraphrasing System for Transforming Regular Expressions into Honorifics
Dongli Han, Shuntaro Kamochi, Xin Song, Naoki Akegawa and Tomomasa Hori 1

PyCWN: a Python Module for Chinese Wordnet
Yueh-Cheng Wu and Shu-Kai Hsieh 5

Annotation Tool for Discourse in PDT
Jiří Mírovský, Lucie Mladová and Zdeněk Žabokrtský 9

LTP: A Chinese Language Technology Platform
Wanxiang Che, Zhenghua Li and Ting Liu 13

Have2eat: a Restaurant Finder with Review Summarization for Mobile Phones
Giuseppe Fabbrizio, Narendra Gupta, Sveva Besana and Premkumar Mani 17

COMUNICA - A Question Answering System for Brazilian Portuguese
Rodrigo Wilkens, Aline Villavicencio, Daniel Muller, Leandro Wives, Fabio Silva and Stanley Loh 21

YanFa: An Online Automatic Scoring and Intelligent Feedback System of Student English-Chinese Translation
Yan Tian 25

HCAMiner: Mining Concept Associations for Knowledge Discovery through Concept Chain Queries
Wei Jin and Xin Wu 29

A High-Performance Syntactic and Semantic Dependency Parser
Björkelund Anders, Bohnet Bernd, Love Hafdell and Pierre Nugues 33

PanLex and LEXTRACT: Translating all Words of all Languages of the World
Timothy Baldwin, Jonathan Pool and Susan Colowick 37

Antelogue: Pronoun Resolution for Text and Dialogue
Eleni Miltsakaki 41

E-HowNet and Automatic Construction of a Lexical Ontology
Wei-Te Chen, Su-Chu Lin, Shu-Ling Huang, You-Shan Chung and Keh-Jiann Chen 45

Cloud Computing for Linguists
Dorothee Beermann and Pavel Mihaylov 49

HowNet and Its Computation of Meaning
Zhendong Dong, Qiang Dong and Changling Hao 53

Multiword Expressions in the wild? The mwetoolkit comes in handy
Carlos Ramisch, Aline Villavicencio and Christian Boitet 57

A Paraphrasing System for Transforming Regular Expressions into Honorifics

Dongli Han, Shuntaro Kamochi, Xin Song, Naoki Akegawa, Tomomasa Hori
Department of Computer Science and System Analysis,
College of Humanities and Sciences, Nihon University
han@cssa.chs.nihon-u.ac.jp

Abstract

Honorifics in Japanese plays an incredibly important role in all walks of social life. The demand to transform regular expressions in Japanese into honorifics automatically has increased rapidly especially in business situations. This paper reviews existing studies and proposes a system to fill this demand with more practicable functions. The experiment shows the effectiveness of our strategy.

1 Introduction

The Japanese language is a kind of highly specific language in establishing hierarchical relations among people, or paying respects to people comparing with other languages. The honorifics in Japanese include different levels of respectful, humble, and polite speeches which are frequently used in various social or business situations. The mechanism of honorifics in Japanese is so complicated that recent young generations in Japan could hardly master it or use it properly.

This situation has encouraged the study dealing with honorifics in Japanese including automatic paraphrasing. For instance, Noguchi et al. generate all kinds of honorific forms for single verbs automatically (Noguchi et al. 2007). In their study, verbs are considered exclusively, and hence no contextual information has been employed.

In another study, Tazoe et al. have proposed a computer model to translate regular expressions into respectful speeches (Tazoe et al. 2005). They determine the type and level of honorifics for a verb in a sentence based on the subject of the sentence and the listener level, the situation level, and the topic level retrieved from the entire article. Comparing with the study of Noguchi et al., this one is more practical. However, there exist some problems in this work. No strategy seems to have been prepared in case multiple verbs with different agents appear in a same sentence. Another problem is the omission of subjects in Japanese sentences. This will obstruct the determination of the honorific form of the verb. Worst of all, the method proposed in this work seems to be remaining as a computer model without being implemented at all.

This paper describes a practical system developed to transform regular expressions in Japanese into honorific forms automatically. Specifically, we manage to retrieve the hierarchical relationships among characters in a sentence so as to determine different honorific forms for multiple verbs with different agents in the same sentence. Another major difference from previous studies is that we employ a series of strategies to cope with the problem of subject-omission. Here in our study, we mainly concentrate our attention on the situation of composing business e-mails.

We first describe the framework of our system in Section 2, and then some main modules in the following sections. Finally we discuss the experiment and conclude this paper in Section 7.

2 Framework of Our System

Our system contains four main parts: Information-retrieval Unit, Subject-complement Unit, Honorific-Form-Determination Unit, and Paraphrasing Unit. We illustrate the whole framework in Figure 1.

Coling 2010: Demonstration Volume, pages 1–4,
Beijing, August 2010

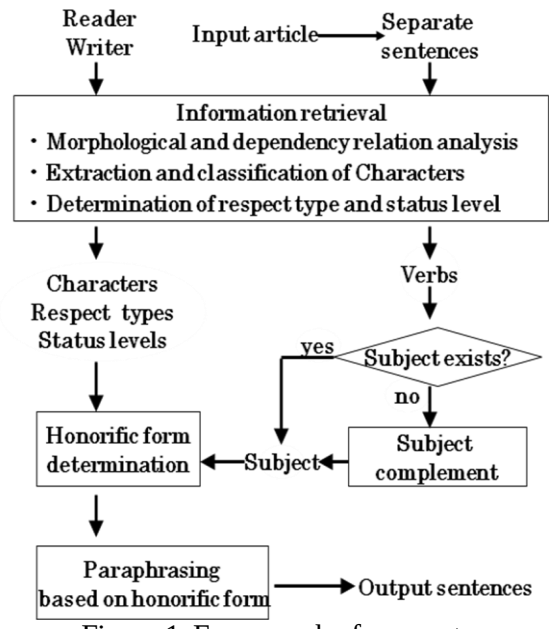


Figure 1: Framework of our system

Before running the system, the user (the person who has composed an e-mail and wants to check the honorifics with the system) is recommended to input the names and the positions or statuses of both himself and the person he is going to contact by e-mail, represented as the *Writer* and the *Reader* in Figure 1. This optional function is to help the system make more precise judgment on the hierarchical relations among characters in the e-mail article, and hence make more reasonable decision on respect type and status level which will be used in Honorific-form-determination Unit.

The procedure will be repeated until all sentences in the input article are processed. We describe the main parts next in section 3, 4, 5, and 6 in detail.

3 Information Retrieval

Information-retrieval Unit is the first and most essential part in our system. We first retrieve basic information including parts of speech and dependency relations among constituent words from a sentence through a free morphological and dependency parsing software, Cabocha[1].

Then based on the basic information obtained above, the system attempts to extract nouns or pronouns representing characters from the sentence, using a Japanese concept thesaurus: EDR Concept Dictionary[2]. The extracted nouns will be divided into three categories: first-person group, second-person group, and third-person group, by checking them against a first-person noun list and a second-person noun list we have made beforehand. The identification results will be used later in the Honorific-form-determination Unit.

Finally, the system assigns a respect type and a status level to each character that is appearing together with nouns showing duty positions or social statuses. Respect type reflects the degree of respect. A larger number indicates a character with higher position, suggesting that a higher honorific form with more regard to the character should be appropriate. Status level has a similar nuance with respect type. It breaks each respect type down into several positions and ranks them according to tiny difference among them.

4 Subject Complement

The system could not determine which kind of honorific form should be applied to a verb with the information on characters only. We need to know the subject of each verb as well. Generally, the subject of a predicate is identified through the dependency analyzing process. However, in case the subject of a verb is omitted, we have to find the subject to help determine the honorific form as described later in Section 5.

In our system, we employ five factors to help recognize the subject for a verb. In this section, we first explain the factors and then describe the method for complementing subjects based on the five factors.

4.1 Nonhuman-behavior Verbs

Our final purpose with this system is to transform a verb into an appropriate kind of honorific form to show the writer's regard or respect to the reader. Situation will be different when the subject of a verb is not a person or character. No respect is needed to be paid to a thing.

In our system, before supplementing the subject, we check the verb against the EDR

[1] http://chasen.org/~taku/software/cabocha/

[2] http://www2.nict.go.jp/r/r312/EDR/J_index.html

Dictionary to see whether the verb represents a nonhuman-behavior. For example, the Japanese verb "沸騰する" meaning *boil*, will never appear with a person as its subject. In this case, the system will not supplement the subject for the verb, but leave a check-mark here to change the verb into a polite speech later in Section 6.

4.2 Expressions for Estimation

There are a number of expressions in Japanese following verbs and implying estimation or hearsay. For example, "だろう" or "でしょう" indicates possibility but uncertainty. In case one of these expressions appears following a verb in a subject-omitted sentence, the subject of the verb tends to be the second person or the third person. We prepare a list containing these expressions and supplement the subject as non-first-person if we find such an expression after the verb.

4.3 Auxiliary Predicates

Expressions following the predicate and appearing in the end of a sentence are called as auxiliary predicates (Kudo et al. 1993). They help predicates describe the modality of a sentence, and at the same time contain the information on subjects. For instance, "~(し)たい" represents the desire of the writer, while "~ても構わない" meaning that *it is all right (for somebody) to do (something)*, indicates that *somebody* here should be the second person or the third person.

4.4 Expressions of Internal Feeling

Internal feeling means the emotion or feeling in the back of one's mind, implying that no one could understand or represent your feeling except yourself. In Japanese, we use adjectives or adjective verbs to express internal feelings, and the second person or the third person will never act as the subject of such an adjective or adjective verb. Here is an example. "楽しい" meaning *happy*, is a frequently used adjective. But different from *happy* which can be used for anybody, "楽しい" in Japanese is used only for the first person.

This fact helps us supplement the subject in a sentence with the first-person noun that we have extracted in the Information-retrieval Unit. We use a Japanese lexicon, Goi-Taikei (Ikehara et al. 1999) as the data source, and employ all the adjectives or adjective verbs in the category of *state of mind of a person* in our system.

4.5 Property of Case

In most situations, if a character or person noun has been used as a surface case with some certain particles in a sentence, the character will seldom act as other surface cases in the same sentence (Isozaki et al. 2006). Along this idea, we avoid supplementing subjects with non-first-person characters or person nouns if they have appeared as other surface cases. Here the reason we exclude the first-person characters from applying the rule lies in the fact that some first person characters do act as multiple surface cases although not that frequently.

4.6 Subject-complement Procedure

Our system tries to supplement the subject for a verb in a sentence utilizing all the previously described factors in a comprehensive manner. At first, every rule is checked to see whether it is applicable or not. Then we generate a slot containing four bits representing nonhuman, the first person, the second person, and the third person respectively for each rule.

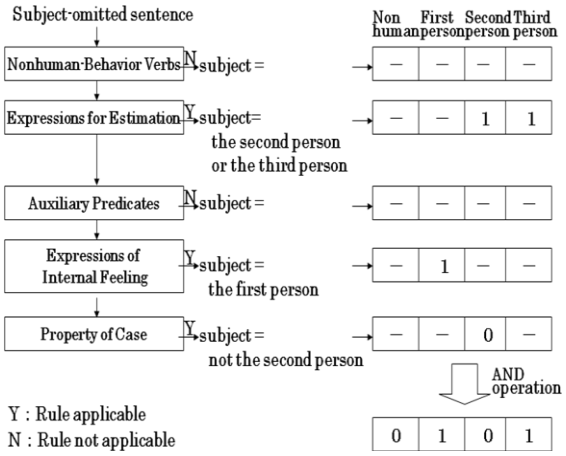


Figure 2: An example of subject complement

According to the applying result of each rule, each slot is updated with 1 or 0 representing possibility and impossibility at the appropriate bit. At last, we carry out an And Operation

with all slots and get the final answer. Figure 2 is an example of subject complement.

If we get multiple candidates for the omitted subject, we have to determine the final one based on the priority order: nonhuman > the first person > the second person > the third person as shown in Figure 2. We have established the above priority order from the result of a preliminary experiment.

Here in this example, the system will supplement the subject of the corresponding verb in the sentence with the first-person noun.

5 Honorific Form Determination

In this section, we describe the method of determining the honorific forms for verbs. We have obtained the respect types, the status levels, and have supplemented the subjects for verbs in Information-retrieval Unit and Subject-complement Unit respectively. Now, the system will determine the honorific form for each verb according to the following rules (R1~R4). Here, the signals *sub*, n^{th} *P*, and SL_n indicates the subject, the nth person, and the status level of the nth person.

R1. If ((*sub* = 2^{nd} *P*) and ($SL_1 < SL_2$)) or
((*sub* = 3^{rd} *P*) and ($SL_1 < SL_3$) and ($SL_2 < SL_3$))
Then Respectful Speech
R2. If ((*sub* = 1st *P*) and ($SL_1 < SL_2$))
Then Humble Speech
R3. If ((*sub* = 1st *P*) or (*sub* = 3^{rd} *P*)) and ($SL_1 < SL_2$) and ($SL_3 < SL_2$)
Then Teichogo Speech
R4. Otherwise Polite Speech

The formula $SL_m < SL_n$ means that the nth person has a higher position than the mth person.

6 Paraphrasing

In accordance with the results of honorific-form determination, we transform verbs in each sentence into their corresponding speeches. There are two types of transformation. One is with most normal verbs based on general paraphrasing rules and the respect levels that we have got in Section 3, such as the verb "働く" meaning *work*, and "書く" meaning *write*.

Another transformation is more complicated. We have to convert the original verb into some particular form first, and then inflect the new form according to the same general paraphrasing rules as those being used for normal verbs. Here is an example. The verb "行く" meaning *go*, holds a particular form: "いらっしゃる" for expressing respect, and "参る" for expressing modesty.

Besides, we have added some exception processing into our system to cope with individual or isolated cases.

7 Conclusions

We have conducted a questionnaire to examine the practicality of our system. Participants in the questionnaires include 5 Japanese college students. They are told to evaluate the naturalness and correctness of a set of transformed articles from our system in 3 levels: 2 for good, 0 for bad, and 1 for the intermediate level between good and bad: not good but acceptable. The average evaluation result is 1.32 showing the effectiveness of our system. We believe that the system could be utilized in situations of creating business documents or learning honorifics in Japanese.

References

Ikehara, S., Miyazaki, M., Shirai, S., Yokoo, A., Nakaiwa, H., Ogura, K., Ooyama, Y., and Hayashi, Y. 1999. *Goi-Taikei - A Japanese Lexicon*, Iwanami Shoten, Tokyo. (in Japanese)

Isozaki, H., Kazawa, H., and Hirao, T. 2006. Japanese Zero Pronoun Resolution Based on Lexicographical Ordering of Penalties. *IPSJ Trans.* 47(7):2279-2294. (in Japanese)

Kudo, I., and Tomokiyo, M. 1993. An Ellipsis-Resolution Mechanism by Using Japanese Predicate Particularity. *IEICE Trans.* J76-D-II(3):624-635. (in Japanese)

Noguchi, S., Nanjo. H., and Yoshimi, T. 2007. Doushi No Tsujohyogen Kara Keigohyogen Eno Kangen. *Proc. of the 13th Annual Meeting of the Association for Natural Language Processing*, pages 978-981. (in Japanese)

Tazoe, T., Watanabe, C., Shiino, T. 2005. Development of a Computer Model for Translating in Respect Language. *IPSJ SIG Notes* 2005(94):1-6. (in Japanese).

PyCWN: a Python Module for Chinese Wordnet

Yueh-Cheng Wu
Institute of Linguistics
Academia Sinica, Taiwan
wyc.juju@gmail.com

Shu-Kai Hsieh
National Taiwan Normal University /
Academia Sinica, Taiwan
shukai@gmail.com

Abstract

This presentation introduces a Python module (PyCWN) for accessing and processing Chinese lexical resources. In particular, our focus is put on the Chinese Wordnet (CWN) that has been developed and released by CWN group at Academia Sinica. PyCWN provides the access to Chinese Wordnet (sense and relation data) under the Python environment. The presenation further demonstrates how this module applies to a variety of lexical processing tasks as well as the potentials for multilingual lexical processing.

1 Introduction

In the presentation, we demonstrate a useful python module for the processing of Chinese lexical semantic resources, viz Chinese Wordnet (CWN). This tool is one of a series of computational processing modules that we have been developing, for a variety of Chinese computational lexical semantic tasks, such as Word sense disambiguation (WSD), Word sense induction (WSI), Automatic relations discovery, etc.

Based on the OOP paradigm, this module enables a programmer to handle CWN synsets and lexical relations in a more efficient way. Written in the python language, it can be run on a broad range of platforms and with the advantages of being able to be imported into other large-scale freely available NLP modules (e.g. Natural Language Processing Toolkits (NLTK)) for advanced surveys.

2 Python Modules for WordNet Processing

Inspired by psycholinguistic theories of human lexical memory, WordNet (Miller et al, 1993) has been considered to be an important lexical resource for both theoretical and computational linguistics. It is organized as a lexical network which centers on synsets (synonymous sets), and the lexical semantic relations (hyponymy, meronymy, etc) are intertwined with the synsets.

The growing amount of studies and applications carried out on wordnets has led to the worldwide efforts in constructing wordnets of different languages, with the envisioned framework of Global Wordnet Grid.[1] To make good use of these wordnet data, an amount of browsers have been proposed. However, it is soon realized that WordNet browsers are not suitable for scaled computational experiments. And ad-hoc processing scripts developed separately without any collaboration and shared architecture did not ease the tasks in the research community.

Later on, an open source python library called the *Natural Language Toolkits* (NLTK) (Bird et al. 2009) has been implemented and distributed. NLTK is designed with many rationales in mind, such as *extensibility*, *modularity*, etc. In NLTK, a **WordNetCorpusReader**, which contains classes and methods for retrieval of sense and relation data, and the calculation of semantic similarity, is designed for accessing Princeton wordnet or its variants

Despite the fact that these tremendous works do help much in accessing wordnet data, in applying to Chinese Wordnet, we found that an extended re-implementation of the module is necessary due to the particularity of the CWN architecture, which will be elaborated on later.

3 PyCWN: Python Modules for Chinese Lexical Ontology

Coling 2010: Demonstration Volume, pages 5–8,
Beijing, August 2010

3.1 Chinese Wordnet

The construction of Chinese Wordnet developed by Academia Sinica follows two lines of thought: (i) multilingual wordnets bootstrapping approach (cf. Sinica BOW[2]), and (ii) linguistically oriented analysis from scratch (cf. CWN[3]). Both of them can be merged organically. In this paper, we focus only on the CWN part.

Generally speaking, NLTK WordnetCorpusReader cannot be seamlessly applied to CWN with the following reasons:

- Distinction of Sense and **Meaning Facet:** CWN proposed that lexical polysemy can be distinguished into two levels: senses and meaning facets (Ahrens et al. 1998). These two levels of polysemies result in a special design for synset.
- Labeling of **Paronymy:** CWN defines paronymy as the relation between any two lexical items belonging to the same semantic classification. (Huang et al, 2007), and label the relation among senses instead of synsets.
- Distinction of Synonyms and **Chinese Written Variants**: CWN regards synonyms and variants differently. Variants are the corresponding words/characters that have different written forms but the same meaning and the identical pronunciation as the target word. In PyCWN, the variants are integrated into the synset of the target word. No new category is created.
- **Homographic Variants**: Homographic variants are the words with same graph but unrelated meanings. CWN defines them as different lemmas. For instance, 連(lian2) has three lemmas. In PyCWN, there is no Lemma class, but the lemma information is retained in the identifier of a synset/sense/meaning facet.

3.2 Architecture of PyCWN

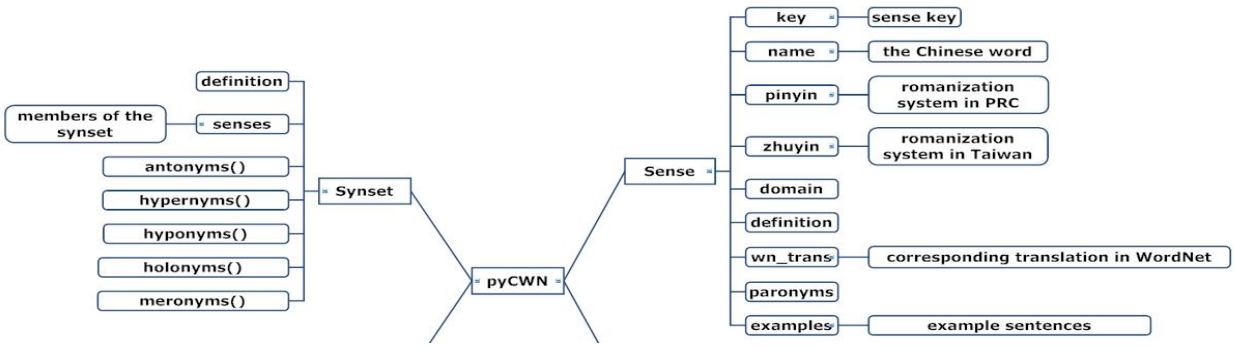


Figure 1: Main structure of PyCWN

Classes in PyCWN follow the main structure of the Chinese Wordnet. Therefore, paronyms are defined between two lexical items while other semantic relations are shared within the same synset. Every member within a synset is a sense or a meaning facet. The Facet class has all the properties as Sense class, and hence is not shown above. The identifier form in CWN is *word(reference_id)*, but for the incorporation to other wordnets, the identifier form in PyCWN is adjusted to be *word.pos.reference_id*.

3.3 Demo

For the reusability of the information extracted, all information is extracted as a string or a list. And because of the coding, Chinese words are not readable in lists. In order to read the result, 'print' is needed. The following figure is an example of the Synset and the Sense class. The Facet class has the same properties as Sense class.

```
>>> import cwn
>>> cwn.synsets('朝代')
['\xe6\x9c\x9d\xe4\xbb\xa3.n.0100']
>>> cwn.synsets('不加')
['\xe4\xb8\x8d\xe5\x8a\xa0.d.0100']
>>> print cwn.synsets('朝代')[0], cwn.synsets('不加')[0]
朝代.n.0100 不加.d.0100
>>> dynasty = cwn.Synset(cwn.synsets('朝代')[0])
>>> bu4jia1 = cwn.Synset(cwn.synsets('不加')[0])
>>> for member in dynasty.senses: # members of the synset
        print member


代1.n.0710
朝1.n.0510
>>> for member in bu4jia1.hypernyms():
        print member


不1.d.0110
勿.d.0200
無1.d.0310
>>> for member in dynasty.meronyms():
        print member


代1.n.0810
朝1.n.0610

>>> dy_sense = cwn.Sense(cwn.synsets('朝代')[0])
>>> print dy_sense.key, dy_sense.name, dy_sense.pinyin, dy_sense.zhuyin
04164201 朝代 chao2 dai4 ㄔㄠˊ ㄉㄞˋ
>>> dy_sense.wn_trans #WordNet translation
[u'dynasty.05976600N..']
>>> for eg in dy_sense.examples:
        print eg


任何一個文明或<朝代>，當時局有了變化，挑戰的力量即刻出現。
我國早在春秋戰國時代就有楚材晉用，而後每到國力發皇的<朝代>，常有延攬外國人擔任要職。
越是興盛強大的<朝代>，對人民性事的管束越是寬鬆；越是衰敗的<朝代>，對人民的控制越緊，性的禁錮也就越厲害。
```

Figure 2: The illustrations of Class methods and Sense properties.

3.4 Cross-linguistic Lexical Studies with NLTK Wordnet Modules

Since the synsets in CWN are already mapped to those in Princeton WordNet via lexical relations, it is easy to perform cross-linguistic lexical comparative studies given the fact that Princeton WordNet is also connected with other wordnets such as EuroWordnet. For example, the following figure shows that *達(da2)* has a hyponym -- *到(dao4)*, and that the WordNet synset *reach.01369399V* is a hypernym(上位詞) of *達(da2)*. Thus it is inferred that *reach.01369399V* should be a hypernym of 到 (dao4) as well. And the information extracted has confirmed this point of view.

```
>>> cwn.synsets('達')
['\xe9\x81\x941.n.0710', '\xe9\x81\x941.v.0110', '\xe9\x81\x941.v.0210', '\xe9\x81\x941.v.0310',
'\xe9\x81\x941.v.0410', '\xe9\x81\x941.v.0510', '\xe9\x81\x941.v.0610', '\xe9\x81\x942.n.0120']
>>> da2 = cwn.synsets('達')[6]
>>> cwn.Synset(da2).hyponyms()
['\xe5\x88\xb0.v.0400']
>>> print cwn.Synset(da2).hyponyms()[0]
到.v.0400
>>> dao4 = cwn.Synset(da2).hyponyms()[0] # dao4 is a hyponym of da2
>>> cwn.Sense(da2).wn_trans
[u'reach.01369399V.\u4e0a\u4f4d\u8a5e.']
>>> print cwn.Sense(da2).wn_trans[0]
reach.01369399V.上位詞.
>>> print cwn.Sense(dao4).wn_trans[0]
reach.01369399V.上位詞.
```

Figure 3: Mapping between CWN and Princeton WordNet

3.5 Availability

The demos will be available as both locally based and remotely accessible from http://lope.eng.ntnu.edu.tw/pycwn/

4 Conclusion

In this presentation, we have demonstrated a python module called PyCWN for the processing of the data in Chinese Wordnet. Now we are also working on the incorporation of NLTK, and extension of the module to a larger Chinese NLP framework, which includes word segmentation and the access of hanzi data, the Gigaword corpus, and the bilingual ontology, etc. We believe that the whole project will be an important infrastructure of Chinese NLP.

References

Ahrens, K., Chang, L., Chen, K., and Huang, C., 1998, Meaning Representation and Meaning Instantiation for Chinese Nominals. *Computational Linguistics and Chinese Language Processing*, 3, 45-60.

Bird, Steven, Ewan Klein and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly.

Huang, Chu-Ren, Shu-Kai Hsieh, Jia-Fei Hong, et al. 2010. Chinese Wordnet: Design, Implementation, and Application of an Infrastructure for Cross-lingual Knowledge Processing. *Zhong Guo YuWen*, 24(2). [in Chinese].

Huang, Chu-Ren, I-Li Su, Pei-Yi Hsiao, and Xiu-Ling Ke. 2007. Paranyms, Co-Hyponyms and Antonyms: Representing Semantic Fields with Lexical Semantic Relations. Chinese Lexical Semantics Workshop. 2007. May 20-23. Hong Kong: Hong Kong Polytechnic University.

[1] http://www.globalwordnet.org/gwa/gwa_grid.htm

[2] http://bow.sinica.edu.tw/

[3] http://cwn.ling.sinica.edu.tw/

Annotation Tool for Discourse in PDT

Jiří Mírovský, Lucie Mladová, Zdeněk Žabokrtský

Charles University in Prague
Institute of Formal and applied Linguistics
{mirovsky,mladova,zabokrtsky}@ufal.mff.cuni.cz

Abstract

We present a tool for annotation of semantic inter-sentential discourse relations on the tectogrammatical layer of the Prague Dependency Treebank (PDT). We present the way of helping the annotators by several useful features implemented in the annotation tool, such as a possibility to combine surface and deep syntactic representation of sentences during the annotation, a possibility to define, display and connect arbitrary groups of nodes, a clause-based compact depiction of trees, etc. For studying differences among parallel annotations, the tool offers a simultaneous depiction of parallel annotations of the data.

1 Introduction

The Prague Dependency Treebank 2.0 (PDT 2.0; Hajič et al., 2006) is a manually annotated corpus of Czech. It belongs to the most complex end elaborate linguistically annotated treebanks in the world. The texts are annotated on three layers of language description: morphological, analytical (which expresses the surface syntactic structure), and tectogrammatical (which expresses the deep syntactic structure). On the tectogrammatical layer, the data consist of almost 50 thousand sentences.

For the future release of PDT, many additional features are planned, coming as results of several projects. Annotation of semantic inter-sentential discourse relations (Mladová et al., 2009)[1] is one of the planned additions. The goal is not only to annotate the data, but also to compare the representation of these relations in the Prague Dependency Treebank with the annotation done at the Penn Treebank, which was carried out at University of Pennsylvania (Prasad et al., 2008).

Manual annotation of data is an expensive and time consuming task. A sophisticated annotation tool can substantially increase the efficiency of the annotations and ensure a higher inter-annotator agreement. We present such a tool.

2 Tree Editor TrEd and the Annotation Extension

The primary format of PDT 2.0 is called PML. It is an abstract XML-based format designed for annotation of linguistic corpora, and especially treebanks. Data in the PML format can be browsed and edited in TrEd, a fully customizable tree editor (Pajas and Štěpánek, 2008).

TrEd is completely written in Perl and can be easily customized to a desired purpose by extensions that are included into the system as modules. In this paper, we describe the main features of an extension that has been implemented for our purposes. The data scheme used in PDT 2.0 has been enriched too, to support the annotation of the discourse relations.

2.1 Features of the Annotation Tool

A tool for the annotation of discourse needs to offer several features:

- creation of a link between arguments of a relation
- exact specification of the arguments of the relation

[1] It is performed in the project *From the structure of a sentence to textual relations* (GA405/09/0729), as one of several tasks.

Coling 2010: Demonstration Volume, pages 9–12,
Beijing, August 2010

- assigning a connective to the relation
- adding additional information to the relation (a type, a source, a comment etc.)

Links between arguments: The annotation of discourse relations in PDT is performed on top of the tectogrammatical (deep syntactic) layer of the treebank. Similarly to another extension of TrEd, dedicated to the annotation of the textual coreference and the bridging anaphora (Mírovský et al., 2010), a discourse relation between nodes is represented by a dedicated attribute at the initial node of the relation, containing a unique identifier of the target node of the relation.[2] Each relation has two arguments and is oriented – one of the arguments is initial, the other one is a target of the link. The link is depicted as a curved arrow between the nodes, see Figure 1. Although the arrow connects the two nodes, it does not mean that the two nodes themselves equal the two arguments of the relation – more about it later.

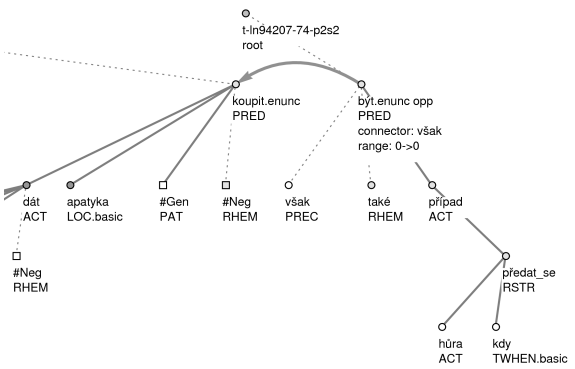


Figure 1. An arrow represents a link.

Additional information about the relation is also kept at the initial node – there is an attribute for the type, an attribute for the source (who annotated it) and an attribute for a comment.

Extent of the arguments: Usually, an argument of a discourse relation corresponds to a subtree of a tectogrammatical tree and can be represented simply by the root node of the subtree. However, there are exceptions to this "rule". Sometimes it is necessary to exclude a part of the subtree of a node from the argument, sometimes the argument consists of more than one tree and sometimes it is even impossible to set exactly the borders of the argument. To allow for all these variants, each discourse link has two additional attributes specifying range of the initial/target argument (both are stored at the initial node of the link). The possible values are:

- "0" (zero) – the argument corresponds to the subtree of the node
- *N* (a positive integer) – the argument consists of the subtree of the node and of *N* subsequent (whole) trees
- "group" – the argument consists of an arbitrary set of nodes (details below); this should only be used if the previous options are not applicable
- "forward" – the argument consists of the subtree of the node and an unspecified number of subsequent trees; should only be used if more specific options are not applicable
- "backward" – similarly, the argument consists of the subtree of the node and an unspecified number of preceding trees; should only be used if more specific options are not applicable

Groups: An argument of a discourse relation can consist of an arbitrary group of nodes, even from several trees. The fact is indicated in a range attribute of the relation (by value "group"). Another attribute then tells which group it is. Groups of nodes inside one document are identified by numbers (positive integers). Each node can be a member of several groups; a list of identifiers of groups a node belongs to is kept at the node. Every group has a representative node – if a discourse link starts/ends at a group, graphically it starts/ends at the representative node of the group, which is the depth-first node of the group belonging to the leftmost tree of the group. Figure 2 shows an example of a group. In the example, the right son (along with its subtree) of the target node of the relation has been excluded from the target argument of the relation (by specifying the target group of nodes, which is graphically highlighted). The right son (and its subtree) is actually the initial argument of the relation.

[2] The data representation allows for several discourse links starting at a single node – there is a list of structured discourse elements representing the individual relations.

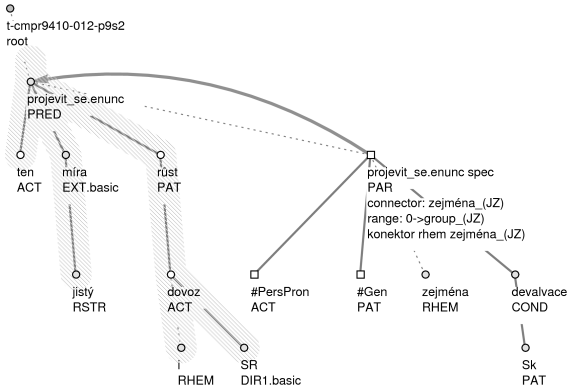


Figure 2. A group of nodes.

Connectives: A connective of a discourse relation is represented as a list of identifiers of (usually) tectogrammatical nodes that correspond to the surface tokens of the connective; the list is kept at the initial node of the relation. It is often only one node, sometimes it consists of several nodes. However, some tokens (like a colon – ":") are not represented on the tectogrammatical layer (at least not as a node). Therefore, identifiers of nodes from the analytical layer are allowed as well.

Collapsed trees: To be able to display more information using less space, a collapsed mode of depicting trees has been implemented.

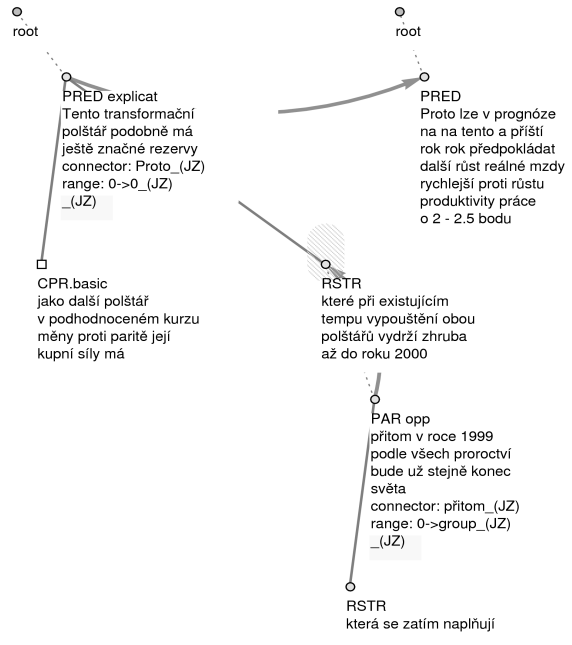


Figure 3. A collapsed mode of depicting trees.

A simple algorithm based on the tectogrammatical annotation has been employed to collapse each subtree representing an individual clause of the sentence into one node. Figure 3 shows an example of two collapsed trees.

Discourse relations most often start/end at nodes representing roots of the clauses. In those rare cases when the discourse relation should lead inside a clause, the annotators can un-collapse the trees, create the link, and collapse back. Such a link would then be depicted with a dotted arrow.

Other features: The tool also incorporates some other features that make the annotation of discourse relations easier. Based on their preference, the annotators can annotate the relations either on the trees or on the linear form of the sentences in the text window of the tool. In the sentences, the tokens that represent the initial/target nodes of the relations are highlighted and easily visible.

2.2 Parallel Annotations

To study discrepancies in parallel annotations, a mode for depicting parallel annotations exists. It can display annotations of the same data from two or more annotators. Figure 4 shows parallel annotations from two annotators. In this example, the two annotators ("JZ" and "PJ") agreed on the relation on the top of the figure, they also marked the same connective ("Poté"), and selected the same type of the relation ("preced(-ence)"). They also agreed on the range of both the arguments ("0", i.e. the subtrees of the nodes). The other relation (on the left, below the first one) has only been recognized by one annotator ("JZ").

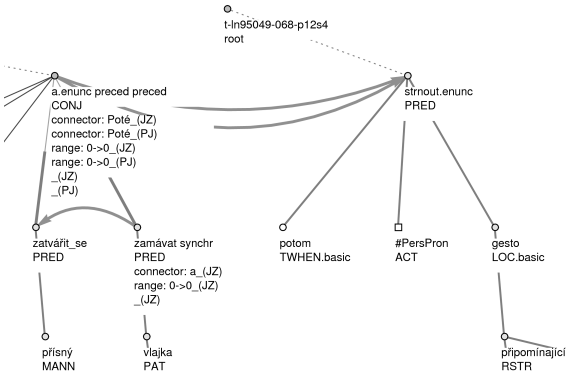


Figure 4. Parallel annotations.

3 Conclusion

From the technical point of view, we have described features of an annotation tool for semantic inter-sentential discourse relations in the Prague Dependency Treebank 2.0. We have shown how it (hopefully in a simple and intuitive manner) allows for quite complex configurations of arguments, and offers features that make the annotation easier. A mode for studying parallel annotations has also been implemented.

Evaluation of such a tool designed for a highly specific task is difficult, as the tool does not produce any direct results (apart from the annotated data) and is highly adapted to our – given the tectogrammatical trees – quite unique needs. (The annotated data themselves, of course, can be (and have been, see Zikánová et al., 2010) evaluated in various ways.) Bird and Liberman (2001) listed some very general requirements on annotation tools for linguistic corpora, namely:

- generality, specificity, simplicity,
- searchability, browsability,
- maintainability and durability.

The first requirement applies both to the annotation tool and the annotation framework. As described e.g. in Mladová et al. (2009), the annotation framework that we use is based on the knowledge obtained from studying various other systems, especially the Penn Discourse Treebank (Prasad et al., 2008), but naturally it has been adjusted to specific needs of the Czech language and PDT. The inter-connection of our system with the tectogrammatical layer of PDT helps in some annotation decisions, as many ambiguities have already been solved in the tectogrammatical annotation.

The second requirement – searchability and browsability – is very easily fulfilled in our framework. A very powerful extension for searching in PML-formatted data, called PML Tree Query, is available in TrEd (Pajas and Štěpánek, 2009).

PML is a well defined formalism that has been used extensively for large variations of data annotation. It can be processed automatically using btred, a command-line tool for applying Perl scripts to PML data, as well as interactively using TrEd. Therefore, we believe that our annotation framework and the annotation tool fulfill also the third requirement.

Acknowledgments

We gratefully acknowledge support from the Czech Ministry of Education (grant MSM-0021620838), and the Grant Agency of the Czech Republic (grants 405/09/0729 and P406/2010/0875).

References

Bird S. and M. Liberman. 2001. *A formal framework for linguistic annotation.* Speech Communication 33, pp. 23–60.

Hajič, J., Panevová, J., Hajičová, E., Sgall, P., Pajas, P., Štěpánek, J., Havelka, J., Mikulová, M., Žabokrtský, Z., and M. Ševčíková-Razímová. 2006. *Prague Dependency Treebank 2.0.* CD-ROM, LDC2006T01, Linguistic Data Consortium, Philadelphia, USA.

Mladová, L., Zikánová, Š., Bedřichová, Z., and E. Hajičová. 2009. *Towards a Discourse Corpus of Czech*. Proceedings of the fifth Corpus Linguistics Conference, Liverpool, UK.

Mírovský, J., Pajas, P., and A. Nedoluzhko. 2010. *Annotation Tool for Extended Textual Coreference and Bridging Anaphora.* Proceedings of LREC 2010, European Language Resources Association, Valletta, Malta.

Pajas, P. and J. Štěpánek. 2008. *Recent advances in a feature-rich framework for treebank annotation.* Proceedings of Coling 2008. Manchester, pp. 673–680.

Pajas, P. and J. Štěpánek. 2009. *System for Querying Syntactically Annotated Corpora.* Proceedings of the ACL-IJCNLP 2009 Software Demonstrations, Association for Computational Linguistics, Suntec, Singapore, pp. 33–36.

Prasad R., Dinesh N., Lee A., Miltsakaki E., Robaldo L., Joshi A., and B. Webber. 2008. *The Penn Discourse Treebank 2.0.* Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC 2008), Marrakech.

Zikánová, Š., Mladová, L., Mírovský, J., and P. Jínová. 2010. *Typical Cases of Annotators' Disagreement in Discourse Annotations in Prague Dependency Treebank*. Proceedings of LREC 2010, European Language Resources Association, Valletta, Malta.

LTP: A Chinese Language Technology Platform

Wanxiang Che, Zhenghua Li, Ting Liu
Research Center for Information Retrieval
MOE-Microsoft Key Laboratory of Natural Language Processing and Speech
School of Computer Science and Technology
Harbin Institute of Technology
{car, lzh, tliu}@ir.hit.edu.cn

Abstract

LTP (Language Technology Platform) is an integrated Chinese processing platform which includes a suite of high performance natural language processing (NLP) modules and relevant corpora. Especially for the syntactic and semantic parsing modules, we achieved good results in some relevant evaluations, such as CoNLL and SemEval. Based on XML internal data representation, users can easily use these modules and corpora by invoking DLL (Dynamic Link Library) or Web service APIs (Application Program Interface), and view the processing results directly by the visualization tool.

1 Introduction

A Chinese natural language processing (NLP) platform always includes lexical analysis (word segmentation, part-of-speech tagging, named entity recognition), syntactic parsing and semantic parsing (word sense disambiguation, semantic role labeling) modules. It is a laborious and time-consuming work for researchers to develop a full NLP platform, especially for Chinese, which has fewer existing NLP tools. Therefore, it should be of particular concern to build an integrated Chinese processing platform. There are some key problems for such a platform: providing high performance language processing modules, integrating these modules smoothly, using processing results conveniently, and showing processing results directly.

LTP (Language Technology Platform), a Chinese processing platform, is built to solve the above mentioned problems. It uses XML to transfer data through modules and provides all sorts of high performance Chinese processing modules, some DLL or Web service APIs, visualization tools, and some relevant corpora.

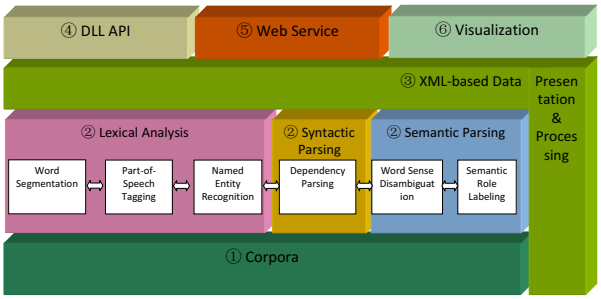


Figure 1: The architecture of LTP

2 Language Technology Platform

LTP (Language Technology Platform)[1] is an integrated Chinese processing platform. Its architecture is shown in Figure 1. From bottom to up, LTP comprises 6 components: ① Corpora, ② Various Chinese processing modules, ③ XML based internal data presentation and processing, ④ DLL API, ⑤ Web service, and ⑥ Visualization tool. In the following sections, we will introduce these components in detail.

2.1 Corpora

Many NLP tasks are based on annotated corpora. We distributed two key corpora used by LTP.

First, WordMap is a Chinese thesaurus which contains 100,093 words. In WordMap, each word sense belongs to a five-level categories. There are 12 top, about 100 second and 1,500 third level, and more fourth and fifth level categories. For instance, the Chinese word "材料" has the following two senses:

[1] http://ir.hit.edu.cn/ltp/

Coling 2010: Demonstration Volume, pages 13–16,
Beijing, August 2010

1. “物(entity) → 统称(common name) → 物资(goods) → 物资(goods) → 材料(material)”

2. “人(human beings) → 才识(ability) → 俊杰(hero) → 人才(talents) → 人才(talents)”

We can see that the two senses belong to “物” (entity) and “人” (human beings) top categories respectively. In each category, the concept becomes more and more specifical.

The second corpus is Chinese Dependency Treebank (CDT) (Liu et al., 2006). It is annotated with the dependency structure and contains 24 dependency relation tags, such as SUB, OBJ, and ADV. It consists of 10,000 sentences randomly extracted from the first six-month corpus of People's Daily (China) in 1998, which has been annotated with lexical tags, including word segmentation, part-of-speech tagging, and named entity recognition tags[2].

2.2 Chinese Processing Modules

We have developed 6 state-of-the-art Chinese processing modules for LTP.

1. Word Segmentation (WordSeg): A CRF model (Lafferty et al., 2001) is used to segment Chinese words. All of the People's Daily (China) corpus is used as training data.

2. Part-of-Speech Tagging (POSTag): We adopt SVMTool[3] for Chinese POS tagging task (Wang et al., 2009). The People's Daily corpus is also used here.

3. Named Entity Recognition (NER): LTP can identify six sorts of named entity: Person, Loc, Org, Time, Date and Quantity. A maximum entropy model (Berger et al., 1996) is adopted here. We still used the People's Daily corpus.

4. Word Sense Disambiguation (WSD): This is an all word WSD system, which labels the WordMap sense of each word. It adopts an SVM model (Guo et al., 2007), which obtains the best performance in SemEval 2009 Task 11: English Lexical Sample Task via English-Chinese Parallel Text.

5. Syntactic Parsing (Parser): Dependency grammar is used in our syntactic parser. A high order graph-based model (Che et al., 2009) is adopted here which achieved the third place of the dependency syntactic parsing subtask in the CoNLL-2009 Syntactic and Semantic Dependencies in Multiple Languages Shared Task (Hajič et al., 2009).

6. Semantic Role Labeling (SRL): SRL is to identify the relations between predicates in a sentence and their associated arguments. The module is based on syntactic parser. A maximum entropy model (Che et al., 2009) is adopted here which achieved the **first place** in the joint task of syntactic and semantic dependencies of the CoNLL-2009 Shared Task.

Modules	Performance	Speed
WordSeg	$F1 = 97.4$	185KB/s
POSTag	The overall $Accuracy = 97.80\%$, and the out of vocabulary word $Accuracy = 85.48\%$	56.3KB/s
NER	The overall $F1 = 92.25$	14.4KB/s
WSD	The all word WSD $Accuracy = 94.34\%$ and the multi-sense word $Accuracy = 91.29\%$	7.2KB/s
Parser	LAS (Labeled Attachment Score) $= 73.91\%$ and UAS (Unlabeled Attachment Score) $= 78.23\%$	0.2KB/s
SRL	$F1 = 77.15$	1.3KB/s

Table 1: The performance and speed for each module.

Table 1 shows the performance and speed of each module in detail. The performances are obtained with n-fold cross-validation method. The speed is gotten on a machine with Xeon 2.0GHz CPU and 4G Memory.

At present, LTP processes these modules with a cascaded mechanism, i.e., some higher-level processing modules depend on other lower-level modules. For example, WSD needs to take the output of POSTag as input; while before POSTag, the document must be processed with WordSeg. LTP can guarantee that the lower-level modules are invoked automatically when invoking higher-level modules.

2.3 LTML

We adopt eXtensible Markup Language (XML) as the internal data presentation for some reasons. First, XML is a simple, flexible text format, and plays an increasingly important role in the ex-

[2]http://icl.pku.edu.cn/icl_res/

[3]http://www.lsi.upc.edu/∼nlp/SVMTool/

change of a wide variety of data on the Web and elsewhere. Second, there exist many powerful and simple XML parsers. With these tools, we can easily and effectively achieve all kinds of operations on XML. Finally, based on XML, we can easily implement visualization with some script languages such as JavaScript.

Based on XML, we have designed a tag-set for NLP platform, named LTML (Language Technology Markup Language). Basically, we regard a word as a unit. The word has attributes such as id, pos, wsd, etc., which indicate the index, part-of-speech, word sense, etc. information of the word. A sentence consists of a word sequence and then a series of sentences compose a paragraph. The semantic role labeling arguments are attached to semantic predicate words. The meaning of each tag and attribute are explained in Table 2.

Tag	Meaning	Attr.	Meaning
<ltml>	Root node		
<doc>	Document level		
<para>	Paragraph in doc	id	Paragraph index in doc
<sent>	Sentence in para	id	Sentence index in paragraph
<word>	Word in sentence	id	Word index in sentence
		cont	Word content
		pos	Part of speech of word
		ne	Named entity type of word
		wsd	Word sense code in WordMap
		parent	Word id of this word depends on in syntax tree
		relate	Syntax relation type
<arg>	Semantic arguments of a word	id	Argument index of this word
		type	Semantic role of this argument
		beg	Beginning word id of this argument
		end	Ending word id of this argument

Table 2: Tags and attributes of LTML

2.4 DLL API

In order to gain the analysis results of LTP, we provide various DLL APIs (implemented in C++ and Python), which can be divided into three classes: I/O operation, module invoking, and result extraction.

1. I/O Operation: Load texts or LTML files and convert them into DOM (Document Object Model); Save DOM to XML files.

2. Module Invoking: Invoke the 6 Chinese processing modules.

3. Result Extraction: Get the results produced by the modules.

Through invoking these APIs, users can accomplish some NLP tasks simply and conveniently. Assuming that we want to get the part-of-speech tags of a document, we can implement it with Python programming language easily as shown in Figure 2.

```
from ltp_interface import *
CreateDOMFromTxt("test.txt") # Load a text
POStag()                     # Invoke POS tagger
for i in range( CountSentenceInDocument() ):
    # Handle each sentence in a document
    word_list = GetWordsFromSentence(i)   # Get words
    pos_list = GetPOSsFromSentence(i)     #   Get POS
......
```

Figure 2: LTP Python API example

However, the DLL API has some shortcomings. First, it only can be used on Microsoft Windows machines. Second, users must download huge model files when LTP is updated. Third, LTP needs a high performance machine to run. All of above problems prevent from its widespread applications.

2.5 Web Service

In recent years, the Internet has become a platform where we can acquire all kinds of services. Users can build their own applications using LTP Web services conveniently. The LTP Web service has the following four advantages:

1. No need to setup LTP system.
2. No need to burden hardware to run LTP.

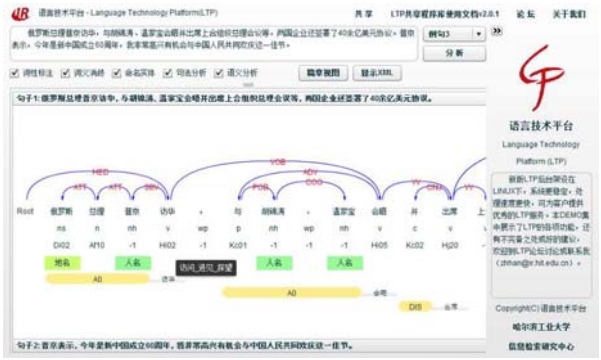


Figure 3: Sentence processing result

3. Update promptly and smoothly.

4. Cross most operating systems and programming languages.

2.6 Visualization

A clear visualization can help researchers to examine processing results. We develop an cross-platform and cross-browser visualization tool with FLEX technology, which can be used easily without installing any excess software.

Figure 3 shows the integrated sentence processing results. The Rows 1 to 4 are the WordSeg, POSTag, WSD, and NER results. The last rows are the SRL results for different predicates. The syntactic dependency Parser tree is shown above with relation labels.

2.7 Sharing

We have been sharing LTP freely for academic purposes[4]. Until now, more than 350 worldwide research institutes have shared LTP with license. Some famous IT corporations of China, such as HuaWei[5] and Kingsoft[6], have bought LTP's commercial license. According to incompletely statistics, there are more than 60 publications which cited LTP, and the LTP web site has more than 30 unique visitors per day on the average.

3 Conclusion and Future Work

In this paper we describe an integrated Chinese processing platform, LTP. Based on XML data presentation, it provides a suite of high performance NLP modules invoked with DLL or Web service APIs, a visualization environment and a set of corpora.

[4]http://ir.hit.edu.cn/demo/ltp/Sharing_Plan.htm

[5]http://www.huawei.com/

[6]http://www.kingsoft.com/

Acknowledgement

This work was supported by National Natural Science Foundation of China (NSFC) via grant 60803093, 60975055, the "863" National High-Tech Research and Development of China via grant 2008AA01Z144, and Natural Scientific Research Innovation Foundation in Harbin Institute of Technology (HIT.NSRIF.2009069).

References

Berger, Adam L., Vincent J. Della Pietra, and Stephen A. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71.

Che, Wanxiang, Zhenghua Li, Yongqiang Li, Yuhang Guo, Bing Qin, and Ting Liu. 2009. Multilingual dependency-based syntactic and semantic parsing. In *CoNLL 2009*, pages 49–54, Boulder, Colorado, June.

Guo, Yuhang, Wanxiang Che, Yuxuan Hu, Wei Zhang, and Ting Liu. 2007. Hit-ir-wsd: A wsd system for english lexical sample task. In *SemEval-2007*, pages 165–168.

Hajič, Jan, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *CoNLL 2009*, pages 1–18, Boulder, Colorado, June.

Lafferty, John, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML 2001*, pages 282–289. Morgan Kaufmann, San Francisco, CA.

Liu, Ting, Jinshan Ma, and Sheng Li. 2006. Building a dependency treebank for improving Chinese parser. *Journal of Chinese Language and Computing*, 16(4):207–224.

Wang, Lijie, Wanxiang Che, and Ting Liu. 2009. An SVMTool-based Chinese POS Tagger. *Journal of Chinese Information Processing*, 23(4):16–22.

Have2eat: a Restaurant Finder with Review Summarization for Mobile Phones

Giuseppe Di Fabbrizio and **Narendra Gupta**
AT&T Labs - Research, Inc.
{pino,ngupta}@research.att.com

Sveva Besana and **Premkumar Mani**
AT&T Interactive - Applied Research
{sbesana,pmani}@attinteractive.com

Abstract

Have2eat is a popular mobile application available for iPhone and Android-based devices that helps users to find and assess nearby restaurants. It lists restaurants located around the device and provides a quick highlight about the opinions expressed by online reviewers. Have2eat summarizes textual reviews by extracting relevant sentences and by automatically generating detailed ratings about specific aspects of the restaurant. A compact one-screen digest allows users to quickly access the information they need, expand to full review pages, and report their experience online by entering ratings and comments.

1 Introduction

Bloggers, professional reviewers, and consumers continuously create opinion-rich web reviews about products and services, with the result that textual reviews are now abundant on the web and often convey a useful overall rating. However, an overall rating cannot express the multiple or conflicting opinions that might be contained in the text and screening the content of a large number of reviews could be a daunting task. For example, a restaurant might receive a great evaluation overall, while the service might be rated below-average due to slow and discourteous wait staff. Pinpointing opinions in documents, and the entities being referenced, would provide a finer-grained sentiment analysis and better summarize users' opinions. In addition, selecting salient sentences from the reviews to textually summarize opinions would add useful details to consumers that are not expressed by numeric ratings. This is especially true for so-called *road warriors* and mobile users "on the run" who are often dealing with limited time and display real estate in searching for a restaurant to make a decision.

Have2eat[1] is a popular[2] mobile application available for iPhone and Android-based devices that addresses these challenges. Have2eat uses the geolocation information either from the GPS device or explicitly entered by the user to produce a list of restaurants sorted by distance and located within a specific radius from the originating location. In addition, when restaurant reviews are available, a compact one-screen digest displays a summary of the reviews posted on the web by other customers. Customers can expand to read a full review page and also enter their own ratings, comments and feedback. The review summaries are visualized on the mobile screen:

- **graphically** by *thumbs-up* (positive reviews) and *thumbs-down* (negative reviews) for different aspects of the restaurant;
- **textually** by a few sentences selected from review texts that best summarize the opinions about various aspects of the restaurant expressed in the reviews;

Extracting opinions from text presents many natural language processing challenges. Prior work on sentiment analysis has been focusing on binary classification of positive and negative opinions (Turney, 2002; Pang et al., 2002; Yu and Hatzivassiloglou, 2003), while aspect rating inference (e.g., the task of determining the opinion polarity in a multi-point scale) has been previously analyzed in Pang and Lee (2005); Goldberg and Zhu (2006); Leung et al. (2006). More recently, Snyder and Barzilay (2007); Shimada and Endo (2008) extended the inference process to multi-aspect ratings where reviews include numerical ratings from mutually dependent aspects. Snyder and Barzilay (2007) shows that modeling the dependencies between aspect ratings in the same reviews helps to reduce the rank-loss (Crammer and Singer, 2001).

[1] www.have2eat.com

[2] More than 400,000 downloads to-date for the iPhone version alone

Coling 2010: Demonstration Volume, pages 17–20,
Beijing, August 2010

There are similar mobile applications obtainable either on the Apple iPhone App Store or as web-based mobile application, such as Zagat[3], UrbanSpoon[4], YP Mobile[5], and Yelp[6], but, to the extent of our knowledge, most of them are only focused on finding the restaurant location based on proximity and some restaurant filtering criterion. When available, restaurant reviews are simply visualized as contiguous list of text snippets with the overall experience rating. None of the listed applications include extended rating predictions and reviews summarization.

2 System Description

The have2eat system architecture is composed of two parts: 1) predictive model training – illustrated in Figure 1 and described in section 2.1, and 2) graphical and textual summarization – shown in Figure 2 and described in section 2.2.

2.1 Graphical summarization by thumbs up/down

The majority of textual reviews available online are accompanied by a single overall rating of the restaurant. To predict consistent ratings for different aspects, namely *food*, *service*, *atmosphere*, *value*, and *overall experience*, we use machine learning techniques to train predictive models, one for each aspect; see Figure 1. More specifically, we used approximately 6,000 restaurant reviews scraped from a restaurant review website[7]. On this website, besides textual reviews, users have also provided numerical ratings for the five aspects mentioned above. Ratings are given on a scale of 1 to 5, 1 being poor and 5 excellent. We experimented with different regression and classification models using a host of syntactic and semantic features. We evaluated these models using rank-loss metrics which measure the average difference between predicted and actual ratings. We found that a maximum entropy (Nigam et al., 1999) model combined with a re-ranking method that keeps in consideration the interdependence among aspect ratings, provided the best predictive model with an average rank-loss of 0.617 (Gupta et al., 2010). This results is better than previous work on the same task as described in Snyder and Barzilay (2007).

To cope with the limited real estate on mobile phones for displaying and allowing users to input their opinions, the predicted ratings were mapped onto thumbs–up and thumbs–down. For each restaurant the proportion of reviews with rating of 1 and 2 was considered thumbs down and ratings of 4 and 5 were mapped to thumbs up. Table 1 shows an example of this mapping.

	Reviews			Thumbs	
	a	b	c	Up	Down
Atmosphere	3	2	4	50%	50%
Food	4	4	5	100%	0
Value	3	2	4	50%	50%
Service	5	5	5	100%	0
Overall	4	4	5	100%	0

Table 1: *Mapping example between ratings and thumbs up/down. Ratings of 3 are considered neutral and ignored in this mapping*

2.2 Textual summaries by sentence selection

Figure 2 shows how summary sentences are selected from textual reviews. As described in the previous section, we trained predictive models for each aspect of the restaurant. To select summary sentences we split the review text into sentences[8]. Using the predictive models and iterating over the restaurant listings, sentences in the reviews are classified by aspect ratings and confidence score. As a result, for each sentence we get 5 ratings and confidence scores for those ratings. We then select a few sentences that have extreme ratings and high confidence and present them as summary text.

We evaluated these summaries using the following metrics.

1. Aspect Accuracy: How well selected sentences represent the aspect they are supposed to.

2. Coverage: How many of the aspects present in the textual reviews are represented in the selected sentences.

[3] mobile.zagat.com
[4] www.urbanspoon.com
[5] m.yp.com
[6] m.yelp.com
[7] www.we8there.com

[8] For this purpose we used a sentence splitter based on statistical models which besides n-grams also uses word part-of-speech as features. This sentence splitter was trained on email data and is 97% accurate.

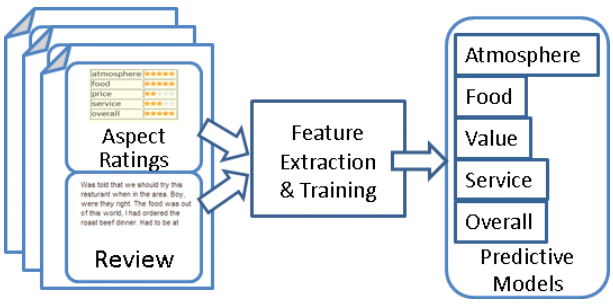


Figure 1: *Predictive model training*

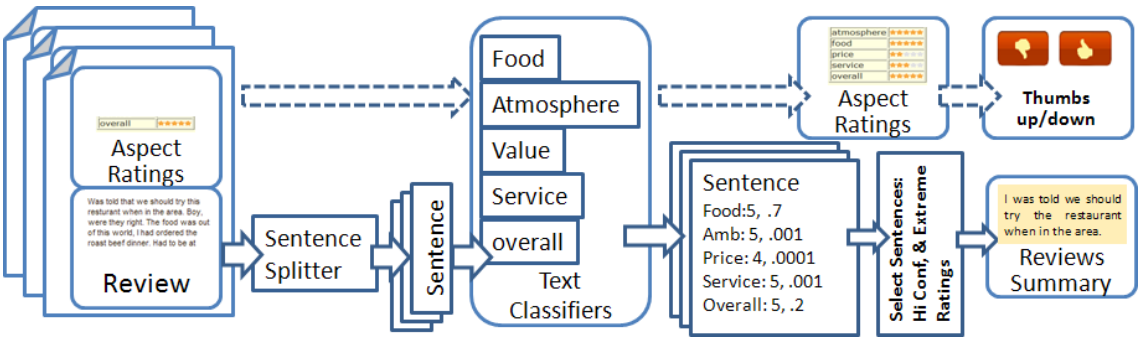


Figure 2: *Graphical and textual summarization*

3. Rating Consistency: How consistent the selected sentences with the summarizing aspect ratings are.

4. Summary quality: Subjective human judgments as to how good the summaries are and automatic multi-document summarization to how good the summaries are compared to a manually created GOLD standard using ROUGE-based (Lin, 2004) metrics.

A detailed description of the summarization task evaluation will be published elsewhere.

3 Demonstration

When launching the application, users are presented with a list of twenty nearby restaurants. The user can browse more restaurants by tapping on a link at the bottom of the page. For each listing we show the distance from the current location and, if available, we provide a thumbs-up or thumbs-down, price information and the summary sentence with the highest confidence score across aspects. Figure 3 shows an example of the *List* page. If users want a list of restaurants for a different location they can tap the *Change* button at the top of the page. This action will bring up the *Location* page where the user can enter city and state and/or a street address.

Users can select a restaurant in the list to view the details, see Figure 4. Details include address, phone number and thumbs up/down for the overall, food, service, value and atmosphere aspects. The user can provide feedback by tapping on the thumbs-up or thumbs-down buttons, as well as by leaving a comment at the bottom of the screen. This page also includes a few summary sentences with extreme ratings and high confidence scores. An example of selected sentences with their polarity is shown in Table 2. By tapping on any of the sentences the users can view the full text of the review from which the sentence was selected. Users can also add a new restaurant by tapping the *Add* icon in the tab bar.

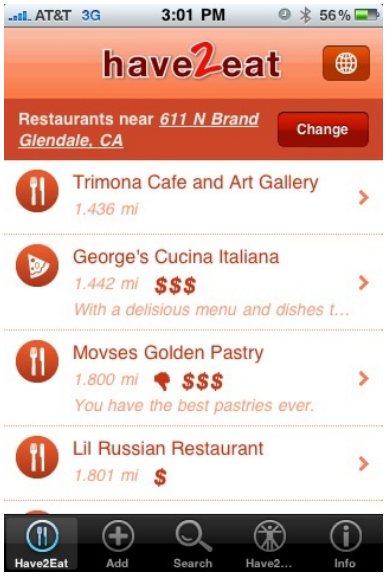


Figure 3: *Have2eat listings screen shot on iPhone*

Figure 5 displays the review selected in the *Details* page along with any other reviews which exist for the restaurant. Users can give feedback on whether they found the review helpful or not by using a thumbs-up or a thumbs-down respectively. Users can also add a review by tapping on a link at the bottom of the page.

4 Conclusion

This demonstration has shown a restaurant finder application for mobile phones, which makes use of summarization techniques to predict aspect ratings from review text and select salient phrases expressing users' opinions about specific restaurant aspects. Users can directly contribute with their feedback by tapping on the aspect thumbs buttons or by directly typing comments.

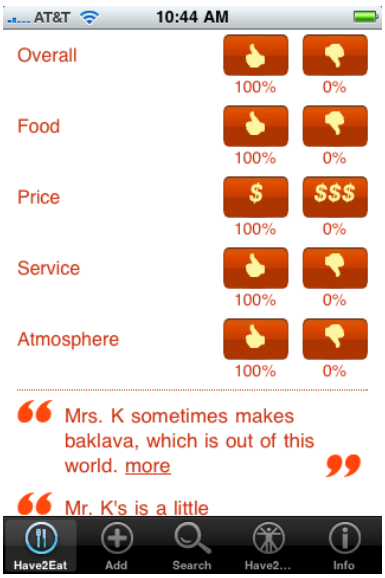


Figure 4: *Have2eat automatically predicted aspect ratings and summary*

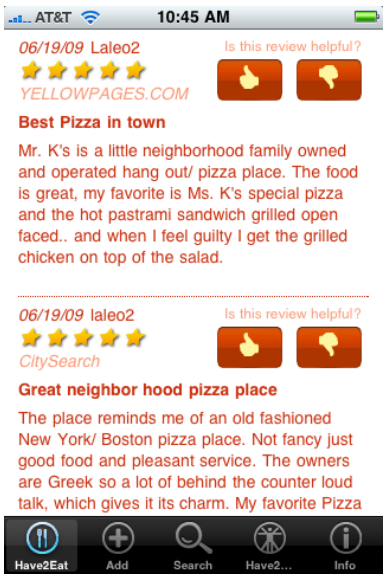


Figure 5: *Have2eat reviews*

Restaurant 1 (3 reviews)	
+	The soups are GREAT! Everything that we have ever ordered has exceeded the ex...
+	Delivery is prompt and credit cards are welcome
+	Their chicken fried rice is the second best in Southern California.
Restaurant 2 (8 reviews)	
+	Great tasting burgers, friendly fast service!
+	The inside is warm and even though the chairs looked uncomfortable, they were not at all.
-	Too many other places to try to worry about getting mediocre food as a high price.
Restaurant 3 (4 reviews)	
+	The salads are tasty, the breadsticks are to die for.
-	We waited approximate 10 more minutes and then asked how much longer.
+	A fun place to go with faimily or a date.
+	If you like salt then this is the place to go, almost everything is full of s...

Table 2: *Example of extracted summaries*

Acknowledgments

We thank Jay Lieske, Kirk Boydston, Amy Li, Gwen Christian, and Remi Zajac for their contributions and great enthusiasm.

References

Crammer, Koby and Yoram Singer. 2001. Pranking with ranking. In Thomas G. Dietterich, Suzanna Becker, and Zoubin Ghahramani, editors, *Neural Information Processing Systems: Natural and Synthetic (NIPS)*. MIT Press, Vancouver, British Columbia, Canada, pages 641–647.

Goldberg, Andrew B. and Jerry Zhu. 2006. Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In *TextGraphs: HLT/NAACL Workshop on Graph-based Algorithms for Natural Language Processing*.

Gupta, Narendra, Giuseppe Di Fabbrizio, and Patrick Haffner. 2010. Capturing the stars: Predicting ratings for service and product reviews. In *Proceedings of the HLT-NAACL Workshop on Semantic Search (SemanticSearch 2010)*. Los Angeles, CA, USA.

Leung, Cane Wing-ki, Stephen Chi-fai Chan, and Fu-lai Chung. 2006. Integrating collaborative filtering and sentiment analysis: A rating inference approach. In *Proceedings of The ECAI 2006 Workshop on Recommender Systems*. Riva del Garda, I, pages 62–66.

Lin, Chin-Yew. 2004. ROUGE: A package for automatic evaluation of summaries. In *Proc. ACL workshop on Text Summarization Branches Out*. page 10.

Nigam, Kamal, John Lafferty, and Andrew Mccallum. 1999. Using maximum entropy for text classification. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*. pages 61–67.

Pang, Bo and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the Association for Computational Linguistics (ACL)*. pages 115–124.

Pang, Bo, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. pages 79–86.

Shimada, Kazutaka and Tsutomu Endo. 2008. Seeing several stars: A rating inference task for a document containing several evaluation criteria. In *Advances in Knowledge Discovery and Data Mining, 12th Pacific-Asia Conference, PAKDD 2008*. Springer, Osaka, Japan, volume 5012 of *Lecture Notes in Computer Science*, pages 1006–1014.

Snyder, Benjamin and Regina Barzilay. 2007. Multiple aspect ranking using the Good Grief algorithm. In *Proceedings of the Joint Human Language Technology/North American Chapter of the ACL Conference (HLT-NAACL)*. pages 300–307.

Turney, Peter. 2002. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the Association for Computational Linguistics (ACL)*. pages 417–424.

Yu, Hong and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

COMUNICA - A Question Answering System for Brazilian Portuguese

Rodrigo Wilkens[♣], Aline Villavicencio[♣], Daniel Muller[♢], Leandro Wives[♣], Fabio da Silva[♠], Stanley Loh[♡]

[♣]Institute of Informatics, Federal University of Rio Grande do Sul (Brazil),
[♢]Conexum (Brazil), [♠]DFL (Brazil), [♡]IntextMining (Brazil)

{rwilkens,avillavicencio,wives}@inf.ufrgs.br, conexum@conexum.com.br, fabio@dfl.psi.br, sloh@terra.com.br

Abstract

COMUNICA is a voice QA system for Brazilian Portuguese with search capabilities for consulting both structured and unstructured datasets. One of the goals of this work is to help address digital inclusion by providing an alternative way to accessing written information, which users can employ regardless of available computational resources or computational literacy.

1 Introduction

A crucial social problem in many countries is functional illiteracy, and in Latin America, according to UNESCO, the process of literacy is only effectively achieved for people who completed at least four years of schooling. Among those who have not completed this cycle of education, there has been high rates of return to illiteracy. According to this definition, in 2002 Brazil had a total of 32.1 million functionally illiterate citizens, representing 26% of the population aged 15 or older[1]. This may have a significant effect on digital inclusion, preventing a considerable part of the population from accessing massive amounts of information such as that available on the Web, or benefitting from advances in technology. Although these figures do not include digital iliteracy, or lack of computational resources, they can give an idea of the magnitude of the problem.

In this context, voice question answering systems (QA) have the potential to make written information more easily accessible to wider audiences as they allow users to ask questions in their own native language and especially if this includes spoken language, sometimes without the need even for a computer (e.g. using the phone). This paper describes COMUNICA, a voice QA system for Brazilian Portuguese with search capabilities for consulting both structured and unstructured datasets. The domain chosen to evaluate the system is that of municipal information from the FAMURS database.[2] One of the goals of this work is to help address digital inclusion by providing a way to overcome (a) difficulties in accessing written information (for visually challenged users), (b) lack of computational resources (for users in remote or computerless areas) and (c) computational illiteracy.

2 QA systems

In recent years, QA has received considerable attention, as can be seen by the initiatives devoted to the task, such as the TREC[3] and CLEF[4]. The task of a QA system is to automatically answer a question in natural language, searching for information in a given data source (e.g. a database, or corpora from a given domain). This is a challenging task as question types can range from lists to facts and definitions, while answers may come from small data sets such as document collections, to the World Wide Web. Moreover, the difficulty of the task is also influenced by whether the questions are restricted to a particular domain (e.g. sports, genes) or not, which additional sources of in-

[1]IBGE: http://www.ibge.gov.br/ibgeteen/pesquisas/educacao.html

[2]http://www.famurs.com.br

[3]http://trec.nist.gov

[4]http://www.clef-campaign.org

Coling 2010: Demonstration Volume, pages 21–24,
Beijing, August 2010

formation are available for a given language (e.g. ontologies of domain-specific knowledge, general ontologies), their coverage, and which tools can be used to help the task (e.g. named entity recognisers, parsers, word sense disambiguation tools). Furthermore, there is no concensus as to the amount of resources and tools that are needed in order to build a working QA system with reasonable performance.

For a resource rich language like English, there is a consistent body of work exemplified by systems such as JAVELIN (Nyberg et al., 2002) and QuALiM (Kaisser, 2005). For other languages, like Portuguese, and particularly the Brazilian variety, QA systems are not as numerous. Over the years, there was an increase in the number of participating systems and data sources in the CLEF evaluation. For instance, in 2004 there were 2 participating systems, and in 2006 it had 4 systems and the best performance was obtained by Priberam (Amaral et al., 2005) with 67% accuracy (Magnini et al., 2006). Figure 1 summarizes the performance of the QA systems for Portuguese for QA@CLEF over the years.

3 COMUNICA Architecture

The Comunica system is composed of five modules: a manager module and four processing modules, as shown in figure 2. The manager is responsible for the integration and communication with the speech recognition, text processing, database access, and speech synthesis modules.

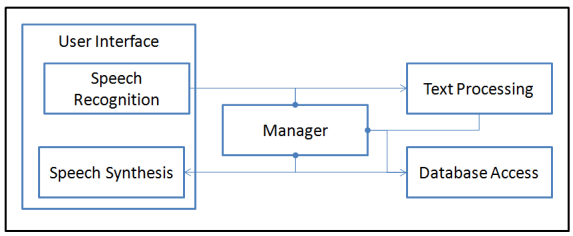


Figure 2: Architecture of the system.

3.1 Speech Recognition

For continuous speech recognition of the users' requests we use an automated phone service. This module uses two research fronts signal analysis (Fourier transform and Wavelets). The coefficients obtained are sequenced on three fronts for continuous speech recognition: HMMs (Becerikli and Oysal, 2007) TDDNN and NESTOR (Nasuto et al., 2009). To train the models, a corpus of FAMURS callcentre telephone interactions has been recorded. The recognition focuses on the vocabulary employed in the domain, in this case municipal information related to taxes from FAMURS. In order to do that, it uses 2 ontologies to validate the candidate words in the input: (a) a general purpose and (b) a domain ontology. The recognised transcribed input is passed to the manager for further processing.

3.2 Text Processing

The manager sends the transcribed input to be processed by the natural language processing module. The natural language queries are processed using shallow and deep tools and accessing both a general and a domain specific ontologies (illustrated in Figure 3). This module needs to determine which type of query the user performed and what is the likely type of answer, based on mostly lexical and syntactic information. This process is divided into 3 mains steps: parsing, concept identification and pattern selection. In the first step, the input is parsed using the PALAVRAS parser (Bick, 2002), and the output provides information about the particular pronoun (wh-word), subject and other verbal complements in the sentence. For concept identification, the system uses the domain ontology, which contains the relevant concepts to be used in next steps. The ontologies also provide additional information about nouns (such as hyperonymy and synonymy) for determining which instances of the concepts were present in the input. For example, "Gramado" is an instance of

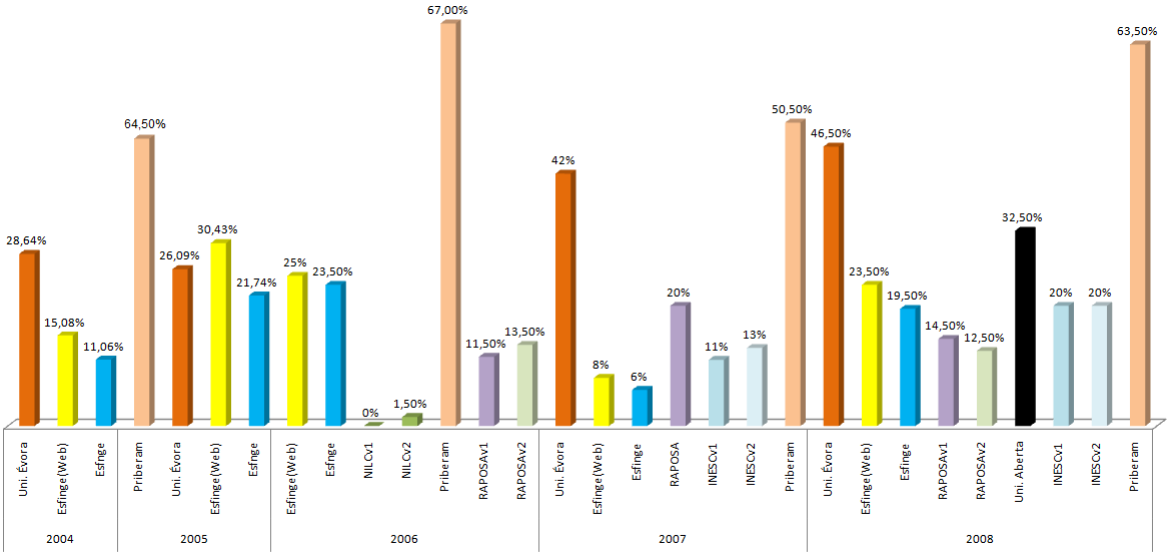


Figure 1: Performance of QA systems for Portuguese QA-CLEF.

the concept "city". Both absolute and relative dates and periods (e.g. last quarter, first week) need to be treated.

Finally, based on this information this module selects from a set of pre-defined question patterns linking concepts of the domain ontology with SQL commands, the one which contains the largest number of concepts in common with the input, and sends it to the manager in an XML format. If there is no complete frame, this module identifies which concepts are missing and returns this in the XML output.

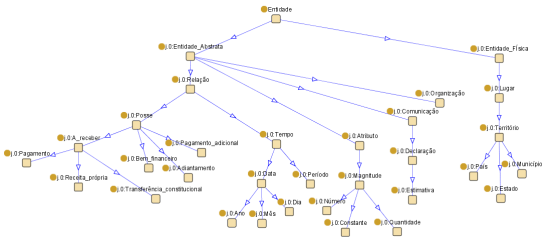


Figure 3: The domain ontology

3.3 Database Access

The search module is divided in two submodules: one for searching information in a structured database and the other for searching in an unstructured knowledge base. It receives as entry an XML file, containing the original input in natural language and the concepts identified in the question. The structured search module receives the input tagged with concepts of the ontology and an identified search pattern, and selects a structured SQL query. These queries are predefined according to the search patterns and the structure of the database. For example, in the case of the FAMURS domain, there are concepts related to time period, cities and taxes. When these 3 concepts are found in the input, a special pattern is selected which defines the kind of information that must be retrieved from the database. An SQL command is then executed in the structured database. All possible patterns are mapped to a specific SQL command. These commands have slots that are filled with instances of the concepts identified in the sentence. For example, names of cities are instances of the concept "city". The retrieved values are used for producing the answer in natural language, using some predefined answer patterns.

Otherwise, the system uses the ADS Digital Company Virtual Assistant (VA) (Duizith et al., 2004) to search the unstructured data (e.g. Frequently Asked Ques-

tions), using the lexical information to locate the answer associated to the most similar question. This answer is written in natural language and will be returned to the main module of the system. If no similar question is found according to a predefined degree of similarity, the VA returns a standard answer.

3.4 Speech Synthesis

The text output to the user is synthesized, resulting in an audio file that is transmitted through the server.

3.5 Manager

The manager is responsible for the integration and communication of the modules. It processes requests, interpreting the actions to be taken and dispatching the requests to specific modules. To start the interaction the manager activates the speech recogniser, and if no problem is detected with the input, it is passed to to the text processing module. In the case of missing information, the manager informs the user that more information is needed. Othwerise, the query is passed to the database module. The database module then returns the result of the query to the manager, which sends this information to the interface component.

All the components are SOA compliant and designed as Web services. This allows us to use a common and simple way of communication among components, allowing a certain degree of independence. Then components can be implemented using different technologies and may be distributed among different servers, if needed.

4 System Demonstration

This is an ongoing project, and a working version of the system will be demonstrated through some text example interactions from the FAMURS domain as the speech recognizer and synthesizer are currently under development. However, users will be able to interact with the other modules, and experience the benefits of natural language interaction for accessing database information.

Acknowledgments

This research was partly supported by CNPq (Projects 479824/2009-6 and 309569/2009-5), FINEP and SEBRAE (COMUNICA project FI- NEP/SEBRAE 1194/07).

References

Amaral, Carlos, Helena Figueira, André F. T. Martins, Afonso Mendes, Pedro Mendes, and Cláudia Pinto. 2005. Priberam's question answering system for portuguese. In Peters, Carol, Fredric C. Gey, Julio Gonzalo, Henning Müller, Gareth J. F. Jones, Michael Kluck, Bernardo Magnini, and Maarten de Rijke, editors, *CLEF*, pages 410–419. Springer.

Becerikli, Yasar and Yusuf Oysal. 2007. Modeling and prediction with a class of time delay dynamic neural networks. *Applied Soft Computing*, 7:1164–1169.

Bick, Eckhard. 2002. *The Parsing System Palavras - Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Famework*. Ph.D. thesis, Aarhus University.

Duizith, José Luiz, Lizandro Kirst da Silva, Daniel Brahm, Gustavo Tagliassuchi, and Stanley Loh. 2004. A virtual assistant for websites. *Revista Eletronica de Sistemas de Informação*, 3.

Kaisser, Michael. 2005. Qualim at trec 2005: Web-question answering with framenet. In *Proceedings of the 2005 Edition of the Text REtrieval Conference*, TREC 2005.

Magnini, Bernardo, Danilo Giampiccolo, Pamela Forner, Christelle Ayache, Valentin Jijkoun, Petya Osenova, Anselmo Peñas, Paulo Rocha, Bogdan Sacaleanu, and Richard F. E. Sutcliffe. 2006. Overview of the clef 2006 multilingual question answering track. In Peters, Carol, Paul Clough, Fredric C. Gey, Jussi Karlgren, Bernardo Magnini, Douglas W. Oard, Maarten de Rijke, and Maximilian Stempfhuber, editors, *CLEF*, volume 4730 of *Lecture Notes in Computer Science*, pages 223–256. Springer.

Nasuto, S.J., J.M. Bishop, and K. DeMeyerc. 2009. Communicating neurons: A connectionist spiking neuron implementation of stochastic diffusion search. *Neurocomputing*, (72):704–712.

Nyberg, Eric, Teruko Mitamura, Jaime G. Carbonell, James P. Callan, Kevyn Collins-Thompson, Krzysztof Czuba, Michael Duggan, Laurie Hiyakumoto, N. Hu, Yifen Huang, Jeongwoo Ko, Lucian Vlad Lita, S. Murtagh, Vasco Pedro, and David Svoboda. 2002. The javelin question-answering system at trec 2002. In *TREC*.

YanFa: An Online Automatic Scoring and Intelligent Feedback System of Student English-Chinese Translation

Yan Tian
School of Foreign Languages
Shanghai Jiao Tong University
tianyan@sjtu.edu.cn

Abstract

Online learning calls for instant assessment and feedback. YanFa is a system developed to score online English-Chinese translation exercises with intelligent feedback for Chinese non-English majors. With the aid of HowNet and Cilin—Chinese Synonym Set (Extended Version), the system adopts the hybrid approach to scoring student translation semantically. It compares student translation with model translation by Synonym Matching, Sentence-pattern Matching and Word Similarity Calculating respectively. The experiment results show that the correlation ratio between the scores given by the system and by human raters is 0.58, which indicates that the algorithm is able to fulfill the task of automated scoring. YanFa is also able to provide feedback on syntactic mistakes made by students through interacting with them. It asks students to analyze the English sentence elements. Then it compares the student analyses with those of the parser and points out the parts which might lead to their wrong understanding as well as their wrong translating.

1 Introduction

Online language learning and instructing are popular in the era of the Internet which calls for instant automated assessment and intelligent feedback. How to provide online translation exercises with immediate scoring and intelligent feedback is a challenging task. Although some researchers (Wang & Chang, 2009; Wen, et al., 2009) have investigated ways to score student translation, they did not aim at fully automated scoring of translation, nor did they try to serve online exercise scoring. Wang & Chang discussed methods of the human-aided automated assessment of translation tests in final exams, and Wen adopted bilingual alignment technology to score translation in language testing. However, online fully automated scoring of translation exercises has its own characteristics. Besides, providing online instant intelligent feedback for students presents another challenge to natural language processing. Up to now very little research, if any, has addressed this topic. In order to meet the demand of online automated scoring of translation exercises and to help students with intelligent feedback, an online automated scoring and intelligent feedback system, called YanFa, has been developed.

This paper aims to outline the framework of YanFa. The paper addresses this by explaining two modules of YanFa, namely, the automatic scoring module and the intelligent feedback

Coling 2010: Demonstration Volume, pages 25–28,
Beijing, August 2010

module. In order to test the accuracy of YanFa, a study with 200 college students was carried out at Shanghai Jiao Tong University. The research intends to verify whether YanFa is able to undertake the task of online automated scoring of student English-Chinese translation as well as the task of providing students with feedback on the mistakes in their comprehending of English sentences, which might lead to their wrong Chinese translation. This paper begins with an introduction, followed by the explanation of the two modules. The experiment is also described. The research findings suggest that YanFa is eligible not only to score student online translation, but also to provide feedback on student syntactic mistakes in their understanding.

2 Automatic Scoring Module

"Translating means translating meaning." (Nida, 1986) Thus, ideally, automated translation scoring should be done at semantic level. Namely, the system should be able to judge whether the student translation is correct in conveying the original meaning to the target language. Therefore, the scoring module should be able to analyze the meaning of student translation which includes word meaning, phrase meaning as well as sentence meaning because translation involves two kinds of transfer: lexical transfer and structural transfer (Hutchins, 1992). Another consideration of building the module is to simulate the manual translation scoring practice in which the sentences are scored according to the correct translation of language points (words and phrases) and that of sentence structures. Usually, 3/4 scores are given to language points and 1/4 to sentence structures.

The automatic scoring module is composed of two parts: the databases and the automatic scoring system. The databases are English Passage Pool, English Sentence Pool, Model Translation Pool, Model Sentence Pattern Pool, Student Translation Pool. The automatic scoring system is composed of a Chinese Parser (SharpICTCLAS.net with precision rate of 97.58 % and recall rate of 90%), a Word Analyzer, a Sentence Pattern Analyzer, a Rater. Besides, Chinese resources, HowNet and Cilin—Chinese Synonym Set (Extended Version by the Lab of Information Retrieval at Harbin Institute of Technology), are also adopted.

First, student translations are parsed by SharpICTCLAS. Then the parsed sentences are sent to Word Analyzer to be compared with the pre-parsed model translations by the same parser. Three different approaches are taken to deal with different parts of speech respectively: nouns are compared with the synonyms in Cilin, of which the seed nouns are from the model translations; verbs, adjectives and adverbs are compared by calculating the word similarity with the aid of HowNet. Similarly, the seed verbs, adjectives and adverbs also come from the model translations. The rest parts of speech, including idioms, are dealt with by key word matching method. After word processing, Sentence Pattern Analyzer compares the sentence patterns of student translations with the model sentence patterns. Last, the results of both analyzers are sent to the Rater which calculates the final score of a student translation. The formulas are as follows:

The formula for Word Analyzer:

Processing of nouns with Cilin:

$$sem\_cl(W_k) = \begin{cases} \dfrac{l}{\lambda\alpha}, & W_k \in C(W_s) \\ 0, & W_k \notin (W_s) \end{cases}$$

where $sem\_cl(W_k)$ refers to the score of a noun in student translation, W_k stands for a noun in student translation, l is the number of parsed parts of speech in model translation, C is the

synonym set of Cilin which embraces the noun appeared in student translation, W_s is a noun in model translation, λ is the total score of the sentences, α is a constant.

Processing of Verbs, Adjectives and Adverbs with HowNet:

$$simhn(W_k) = \arg\max_{1<i<m} (sim(W_i, W_k))$$

where $simhn(W_k)$ is the maximum value of a primitive, W_i is the primitive in HowNet, W_k is a word in student translation, $1<i<m$ means i is bigger than 1, but less than m (m is the number of primitives).

$$sem\_hn(W_k) = \frac{simhn(W_k)}{\lambda\alpha}$$

where $sem\_hn(W_k)$ refers to the score of a word.

Processing of other parts of speech:

$$sem\_st(W_k) = \begin{cases} \dfrac{l}{\lambda\alpha}, & W_k \in T(W_s) \\ 0, & W_k \notin T(W_s) \end{cases}$$

where $sem\_st(W_k)$ means the score of other parts of speech, T refers to the set of other parts of speech.

The formula for Sentence Pattern Analyzer:

$$sim\_pat = \begin{cases} (1-\alpha)\lambda, & AnsTran \in reg(Std\,\mathrm{Re}\,g) \\ 0, & AnsTran \notin reg(Std\,\mathrm{Re}\,g) \end{cases}$$

where sim_pat stands for the score of the sentence pattern of a sentence, $reg(Std\,\mathrm{Re}\,g)$ refers to the set of model translation (standard version) annotated by regular expression, $AnsTran$ means student translation.

The formula for each sentence:

$$score = \alpha\lambda sim\_sem(w_k) + (1-\alpha)\lambda sim\_pat$$

The formula for the total score of a passage (with 5 sentences to be translated as in YanFa system):

$$Totalscore = (\alpha\lambda sim\_sem(W_k) + (1-\alpha)\lambda sim\_pat) \times 5$$

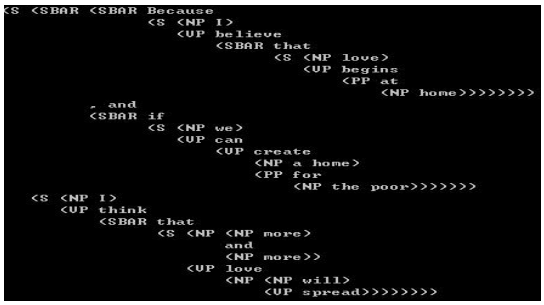
3 Intelligent Feedback Module

It is believed that comprehending of a source language plays a crucial role in its translation, especially when the source language is a foreign language to a translator. Accordingly, correct understanding of English sentences is essential to its translating into Chinese. Therefore, the intelligent feedback module focuses on whether students could correctly understand the English sentences. Specifically, feedback on correct understanding of clauses is provided rather than that of phrases because wrong translation occurs frequently on linguistic units larger than phrases when complex sentences are to be translated by Chinese college students.

The Intelligent Feedback Module is composed of three parts: parsing of the original English sentences, comparing student parsing results with those of the parser, providing feedback to students.

3.1 Parsing

The module employs the English parser by Carnegie Mellon University (free online parser) to parse the original English sentences. It takes the advantage of the "SBAR" sign as the marks of clauses. For example, following is the parsed result of a sentence: "Because I believe that love begins at home and if we can create a home for the poor, I think that more and more love will spread."



3.2 Comparing

The module asks students to mark clauses of the English sentences. Then it compares the marked clauses with the results parsed by the parser through string matching. If the matching fails, the module comes to the decision that a wrong understanding happens.

3.3 Providing feedback

The module is able to provide students with the comparison of their parsing with the right parsing of the whole sentences. If requested, the module is also able to present students with the comparison of their wrongly parsed clauses with the right ones.

4 Experiment

In order to test the accuracy of the automatic scoring system, 200 non-English majors at Shanghai Jiao Tong University were invited to try the online scoring system at "Shanghai Jiao Tong University English Learning Center" (http://english.sjtu.edu.cn). Then the scores were compared with those of two human raters. The correlation ratio is around 0.58 (the correlation ratio between the human raters is 0.67). Also, an online questionnaire was delivered to those who have tried the system to learn their opinions on the scores and the feedback they got. The statistics show that most of the students gave positive responses.

5 Conclusion

YanFa has been developed to score Chinese college students' online English-Chinese translation exercises as well as to provide feedback on their mistakes of understanding the English sentences. Semantic scoring has been explored on lexical level with such resources as HowNet and Cilin. While the scoring on sentence level has to yield to sentence pattern matching due to the unsatisfactory performance of Chinese syntactic parsers. Although this pilot research has achieved its initial purpose, yet it is far from satisfactory. Further efforts should be made in increasing the scoring accuracy and more detailed feedback.

References

Waard, J.D. & Nida, E.A., 1986. *From One Language to Another.* Thomas Nelson Publishers, Tennessee, U.S.A

Wang, L. & Chang, B.B., 2009. *Research on the Human-aided Auto-assessment for Translation Tests in College English.* CAFLEC, No. 128,17-21

Wen, Q.F. et al., *Application of Bilingual Alignment Technology to Automatic Translation Scoring of English Test.* CAFLEC, No. 125, 3-8

Hutchins,W. J. & Somers H. L. ,1992. *An Introduction to Machine Translation*, ACADEMIC PRESS LIMITED, Printed in Great Britain at the University Press, Cambridge. 110-111

D. Callear, J. Jerrams-Smith, and V. Soh, "*CAA of Short Non-MCQ Answers*", Fifth International Computer Assisted Assessment Conference, Loughborough University, 2001.

Brown, H.D.1987. *Principles of Language Learning and Teaching* [M]. Egnlewood Cliffs, NJ: Prentice Hall.

HCAMiner: Mining Concept Associations for Knowledge Discovery through Concept Chain Queries

Wei Jin
Department of Computer Science
North Dakota State University
wei.jin@ndsu.edu

Xin Wu
Department of Computer Science & Technology
University of Science and Technology of China
xinwu@mail.ustc.edu.cn

Abstract

This paper presents *HCAMiner*, a system focusing on detecting how concepts are linked across multiple documents. A traditional search involving, for example, two person names will attempt to find documents mentioning both these individuals. This research focuses on a different interpretation of such a query: what is the best concept chain across multiple documents that connects these individuals? A new robust framework is presented, based on (i) generating concept association graphs, a hybrid content representation, (ii) performing concept chain queries (*CCQ*) to discover candidate chains, and (iii) subsequently ranking chains according to the significance of relationships suggested. These functionalities are implemented using an interactive visualization paradigm which assists users for a better understanding and interpretation of discovered relationships.

1 Introduction

There are potentially valuable nuggets of information hidden in large document collections. Discovering them is important for inferring new knowledge and detecting new trends. Data mining technology is giving us the ability to extract meaningful patterns from large quantities of structured data. Collections of text, however, are not as amenable to data mining. In this demonstration, we describe *HCAMiner*, a text mining system designed to detect hidden information between concepts from large text collections and expose previously unknown logic connections that connect facts, propositions or hypotheses.

In our previous work, we have defined concept chain queries (*CCQ*) (Jin et al , 2007), a special case of text mining in document collections focusing on detecting links between two concepts across text documents. A traditional search involving, for example, two person names will attempt to find documents mentioning both of these names and produce a list of individual pages as result. In the event that there are no pages contain both names, it will return "no pages found" or pages with one of the names ranked by relevancy. Even if two or more interrelated pages contain both names, the existing search engines cannot integrate information into one relevant and meaningful answer. This research focuses on a different interpretation of such a query: what is the best concept chain across documents that potentially connects these two individuals? For example, both may be football lovers, but are mentioned in different documents. This information can only be gleaned from multiple documents. A generalization of this task involves query terms representing general concepts (e.g., airplane crash, foreign policy). The goal of this research is to sift through these extensive document collections and find such hidden links.

Formally, a concept chain query involving concepts A and B has the following meaning: find the most plausible relationship between concept A and concept B assuming that one or more instances of both concepts occur in the corpus, but not necessarily in the same document. We go one step further and require the response to include text snippets extracted from multiple documents in which the discovered relationship

Coling 2010: Demonstration Volume, pages 29–32,
Beijing, August 2010

occurs. This may assist users with the second dimension of the analysis process, i.e., when the user has to peruse the documents to figure out the nature of the relationship underlying a suggested chain.

2 The Proposed Techniques

2.1 The new representation framework

A key part of the solution is the representation framework. What is required is something that supports traditional IR models (such as the vector space model), graph mining and probabilistic graphical models. We have formulated a representation referred to as concept association graphs (*CAG*). Figure 1 illustrates a small portion of *CAG* that has been constructed based on processing the 9/11 commission report[1] in the counterterrorism domain. The inputs for this module are paths for data collection and domain-specific dictionary containing concepts. In our experiments, we extract as concepts all named entities, as well as any noun or noun phrases participating in Subject-Verb-Object relationships. Domain ontological links are also illustrated, e.g., *white house* is a type of *organization*.

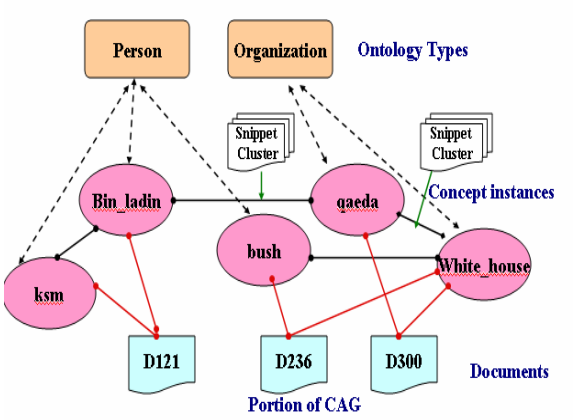


Figure 1. Portion of the CAG

2.2 Concept profile (*CP*) and snippet cluster generation

A concept profile (*CP*) is essentially a set of terms that together represent the corresponding concept. We generate concept profiles by adapting the *Local Context Analysis* technique in Information Retrieval and then integrate them into the graphical framework (Jin et al., 2007).

[1] http://www.9-11commission.gov/

Particularly, the *CP* for concept c is built by first identifying a relevant set of text segments from the corpus in which concept c occurs, and then identifying characteristic concepts from this set and assessing their relative importance as descriptors of concept c. Formally, the profile $Profile(c_i)$ for concept c_i is described by a set of its related concepts c_k as follows:

$$\Pr ofile(c_i) = \{\omega_{i,1}c_1, \omega_{i,2}c_2, \cdots, \omega_{i,k}c_k, \cdots\}$$

Weight $\omega_{i,k}$ denotes the relative importance of c_k as an indicator of concept c_i and is calculated as follows:

$$\omega_{i,k} = \xi + \frac{\log(\ f(i,k) \times idf_k\ )}{\log\ n}$$

Where n is the number of relevant text segments considered for concept c_i (in our experiments, the basic unit of segmentation is a *sentence*). The function $f(i, k)$ quantifies the correlation between concept c_i and concept c_k and is given by

$$f(i,k) = \sum_{j=1}^{n} sf_{i,j} \times sf_{k,j}$$

Where $sf_{i,j}$ is the frequency of concept c_i in the *j-th* sentence and $sf_{k,j}$ is the frequency of concept c_k in the *j-th* sentence. This can be easily computed by constructing "concept by sentence" matrix Q whose entry $Q_{i,j}$ is the number of times concept c_i occurs in sentence s_j. $(QQ^T)_{ij}$ then represents the number of times concepts c_i and c_j co-occur in sentences across the corpus. The inverse document frequency factor is computed as

$$idf_k = \max(\ 1, \frac{\log\ N\ /\ np_k}{\lambda})$$

Where N is the number of sentences in the document collection, np_k is the number of sentences containing concept c_k. λ is a collection dependent parameter (in the experiments λ=3). The factor ζ is a constant parameter which avoids a value equals to zero for $w_{i,k}$ (which is useful, for instance, if the approach is to be used with probabilistic framework). Usually, ζ is a small factor with values close to 0.1. Table 1 illustrates a portion of the *CP* constructed for concept *Bin*

Ladin. The best concepts are shown based on their relative importance.

Table 1. Portion of *CP* for Concept 'Bin Ladin'

Bin Ladin	
Dimension	**Value**
Al-qaeda	0.569744
Afghanistan	0.535689
Sandi Arabia	0.527825
Islamist	0.478891
Islamist Army	0.448877
Extremist	0.413376
Ramzi Yorsef	0.407401
Sudanese	0.370125
Saddam Hussein	0.369928
Covert Action	0.349815
Embassy Bombings	0.313913

Given the information provided by concept profiles, the strength of a relation (edge weight in the *CAG*) between concept c_i and concept c_j is measured by the similarity between their respective profiles. If a concept *X* is related to another concept *Y* which has a similar context as that of *X*, then such a relation can be coherent and meaningful. More precisely, a scalar profile similarity matrix $S_{i,j}$ is defined as follows:

$$S_{i,j} = \frac{\hat{C}(c_i) \bullet \hat{C}(c_j)}{\left|\hat{C}(c_i)\right| \times \left|\hat{C}(c_j)\right|}$$

Where $\hat{C}(c_i)$ and $\hat{C}(c_j)$ are profile vectors for concepts c_i and c_j respectively. In terms of text mining and knowledge discovery, we also require the graphical representation relate concepts and associations to underlying text snippets in the corpus. Without this support, the framework is not complete since users need to validate conclusions by looking at actual documents. This is achieved by associating each edge with a **Snippet Cluster**, which links the snippets (e.g., sentences) in the corpus to the corresponding associations (e.g., co-occurrence of concepts in sentences) represented by edges in the *CAG*. The resulting snippet clusters offer a view of the document collection which is highly characterized by the presence of concept associations (illustrated in Fig. 1).

2.3 Concept Chain Generation and Ranking

Given two concepts of interest designated, *concept chain query* (*CCQ*) tries to find if (i) there is a direct connection (association) between them, or (ii) if they can be connected by several intermediate concepts (paths). Note that finding direct links between two concepts is trivial; in the following we mainly focus on discovering and ranking indirect connections between concepts.

We formulate the *CCQ* problem as finding optimized transitive associations between concepts in the *CAG*. Given the source concept c_1 and destination concept c_n, the transitive strength of a path from c_1 to c_n made up of the links $\{(c_1, c_2), \ldots, (c_{n-1}, c_n)\}$, denoted by $TS(c_1, c_2, \ldots, c_n)$, is given by:

$$TS(c_1, c_2, \cdots, c_n) = \prod_{i=1}^{n-1} (w(c_i, c_{i+1}))$$

Where $w(c_i, c_{i+1})$ represents the weight of the edge connecting concepts c_i and c_{i+1}. The formulation of generating and ranking transitive associations is then described as follows with input and output constraints specified:

Given: an edge-weighted graph *CAG*, vertices *s* and *t* from *CAG*, and an integer budget *l*

Find: ranked lists of concept chains *CCs* staring from *s* and ending at *t*, one list for each possible length (i.e., between the shortest connection length and the specified maximum length *l*). Within each list, top-*K* chains that maximize the "goodness" function $TS(\cdot)$ is returned.

Our optimization problem is now to find an optimal path that maximizes the "goodness" measure for each possible length. This could be easily computed using dynamic programming given the inductive definition of the goodness function $TS(\cdot)$. Notice that in real applications there are often cases that users might be interested in exploring more potential chains instead of just one optimal chain, we have thus adapted the traditional dynamic programming algorithm into finding *top-K* chains connecting concepts for each possible length efficiently. The details of algorithm and implementation can be found in (Jin et al , 2007).

3 The System Interface

Figure 2 illustrates the main *HCAMiner* visualization interface. Given the user specified paths for data collection and domain specific thesaurus,

the *Concept Association Graph* is first constructed. Analyzers are then provided another panel of parameters to guide the discovery process, e.g., *max_len* controls the maximum length of desired chains; *chain_num* specifies the number of top ranked chains to be returned for each possible length. The visualized result for concept chain query involving person names "*Bush*" and "*Bin Ladin*" with parameter values "*max_len*" 3 and "*chain-num*" 5 is shown in Fig. 2. The system offers different views of the generated output:

a) *Chain Solution View* (in the left pane). This view gives the overview of all the generated concept chains.
b) *XML Data View* (in the upper-right pane). This view links each concept chain to the underlying text snippets in the corpus in which the suggested association occurs. Snippets are presented in XML format and indexed by *docId.snippetID*. This makes it easier for analyzers to explore only the relevant snippet information concerning the query.
c) *Concept Profile View*. This view provides the profile information for any concept involved in the generated chains. Figure 2 shows portion of the *CP* generated for Concept 'Bin Ladin' (illustrated on the bottom right).

4 CONCLUSIONS

This paper introduces *HCAMiner*, a system focusing on detecting cross-document links between concepts. Different from traditional search, we interpret such a query as finding the most meaningful concept chains across documents that connect these two concepts. Specifically, the system generates ranked concept chains where the key terms representing significant relationships between concepts are ranked high. The discovered novel but non-obvious cross-document links are the candidates for hypothesis generation, which is a crucial initial step for making discoveries.

We are now researching extensions of concept chains to concept graph queries. This will enable users to quickly generate hypotheses graphs which are specific to a corpus. These matched instances can then be used to look for other, similar scenarios. Ontology guided graph search is another focus of future work.

References

Jin, Wei, Rohini K. Srihari, and Hung Hay Ho. 2007. A Text Mining Model for Hypothesis Generation. *In Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'07)*, pp. 156-162.

Jin, Wei, Rohini K. Srihari, Hung Hay Ho, and Xin Wu. 2007. Improving Knowledge Discovery in Document Collections through Combining Text Retrieval and Link Analysis Techniques. *In Proceedings of the 7th IEEE International Conference on Data Mining (ICDM'07)*, pp. 193-202.

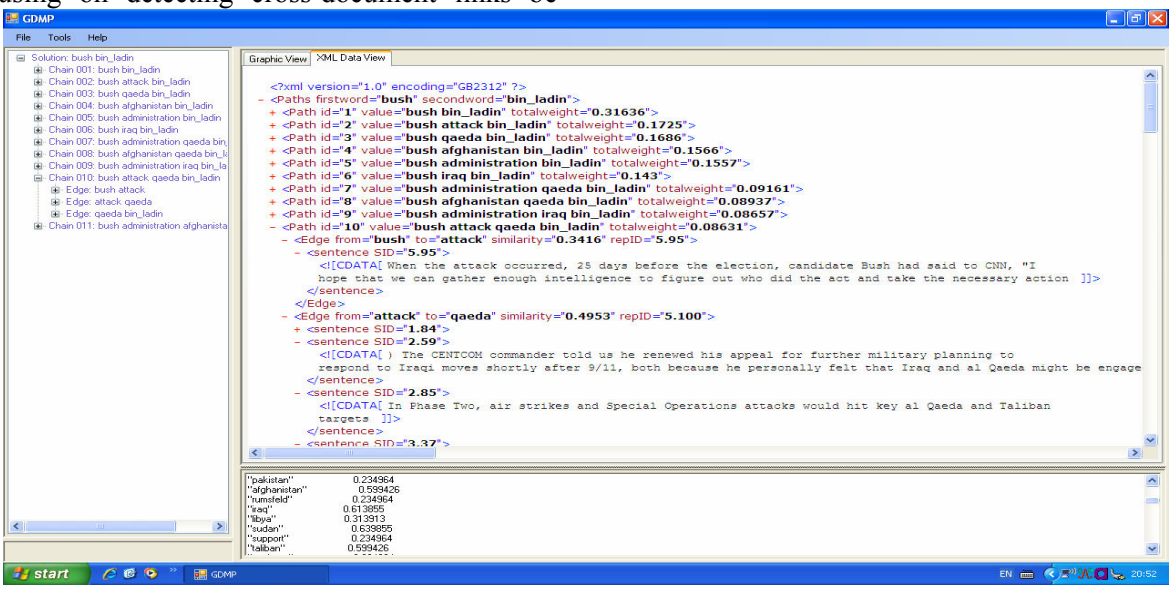


Figure 2. Screenshot of the user interface

A High-Performance Syntactic and Semantic Dependency Parser

Anders Björkelund† **Bernd Bohnet**‡ **Love Hafdell**† **Pierre Nugues**†
†Department of Computer science
Lund University
anders.bjorkelund@cs.lth.se
love.hafdell@cs.lth.se
pierre.nugues@cs.lth.se
‡Institute for Natural Language Processing
University of Stuttgart
bohnet@ims.uni-stuttgart.de

Abstract

This demonstration presents a high-performance syntactic and semantic dependency parser. The system consists of a pipeline of modules that carry out the tokenization, lemmatization, part-of-speech tagging, dependency parsing, and semantic role labeling of a sentence. The system's two main components draw on improved versions of a state-of-the-art dependency parser (Bohnet, 2009) and semantic role labeler (Björkelund et al., 2009) developed independently by the authors.

The system takes a sentence as input and produces a syntactic and semantic annotation using the CoNLL 2009 format. The processing time needed for a sentence typically ranges from 10 to 1000 milliseconds. The predicate–argument structures in the final output are visualized in the form of segments, which are more intuitive for a user.

1 Motivation and Overview

Semantic analyzers consist of processing pipelines to tokenize, lemmatize, tag, and parse sentences, where all the steps are crucial to their overall performance. In practice, however, while code of dependency parsers and semantic role labelers is available, few systems can be run as standalone applications and even fewer with a processing time per sentence that would allow a user interaction, i.e. a system response ranging from 100 to 1000 milliseconds.

This demonstration is a practical semantic parser that takes an English sentence as input and produces syntactic and semantic dependency graphs using the CoNLL 2009 format. It builds on lemmatization and POS tagging preprocessing steps, as well as on two systems, one dealing with syntax and the other with semantic dependencies that reported respectively state-of-the-art results in the CoNLL 2009 shared task (Bohnet, 2009; Björkelund et al., 2009). The complete system architecture is shown in Fig. 1.

The dependency parser is based on Carreras's algorithm (Carreras, 2007) and second order spanning trees. The parser is trained with the margin infused relaxed algorithm (MIRA) (McDonald et al., 2005) and combined with a hash kernel (Shi et al., 2009). In combination with the system's lemmatizer and POS tagger, this parser achieves an average labeled attachment score (LAS) of 89.88 when trained and tested on the English corpus of the CoNLL 2009 shared task (Surdeanu et al., 2008).

The semantic role labeler (SRL) consists of a pipeline of independent, local classifiers that identify the predicates, their senses, the arguments of the predicates, and the argument labels. The SRL module achieves an average labeled semantic F1 of 80.90 when trained and tested on the English corpus of CoNLL 2009 and combined with the system's preprocessing steps and parser.

2 The Demonstration

The demonstration runs as a web application and is available from a server located at `http://`

*Authors are listed in alphabetical order.

Coling 2010: Demonstration Volume, pages 33–36,
Beijing, August 2010

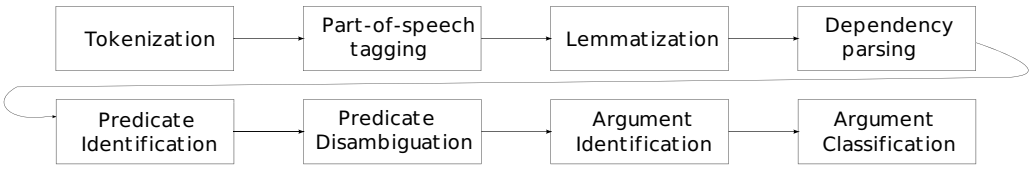


Figure 1: The overall system architecture.

barbar.cs.lth.se:8081/. Figure 2 shows the input window, where the user can write or paste a sentence, here *Speculators are calling for a degree of liquidity that is not there in the market*.

Figure 3 shows the system output. It visualizes the end results as a list of predicates and their respective arguments in the form of colored segments. It also details the analysis as tabulated data using the CoNLL 2009 format (Surdeanu et al., 2008; Hajič et al., 2009), where the columns contain for each word, its form, lemma, POS tag, syntactic head, grammatical function, whether it is a predicate, and, if yes, the predicate sense. Then, columns are appended vertically to the table to identify the arguments of each predicate (one column per predicate). Figure 3 shows that the sentence contains two predicates, *call.03* and *degree.01* and the two last columns of the table show their respective arguments. Clicking on a predicate in the first column shows the description of its arguments in the PropBank or NomBank dictionaries. For *call.03*, this will open a new window that will show that Arg0 is the *demander*, Arg1, the *thing being demanded*, and Arg2, the *demandee*.

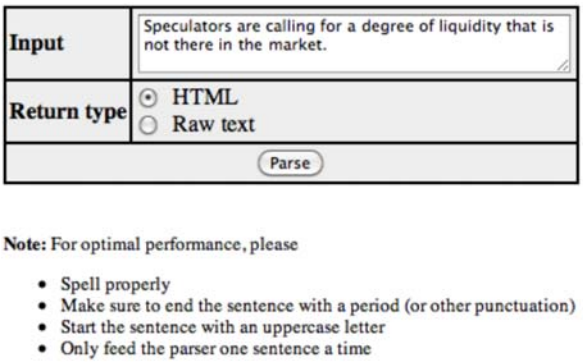


Figure 2: The input window, where the user entered the sentence *Speculators are calling for a degree of liquidity that is not there in the market*. Clicking on the **Parse** button starts the parser.

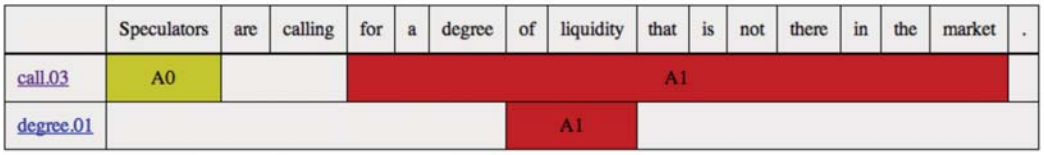
3 Preprocessing Steps

The preprocessing steps consist of the tokenization, lemmatization, and part-of-speech tagging of the input sentence. We use first OpenNLP[1] to tokenize the sentence. Then, the lemmatizer identifies the lemmas for each token and the tagger assigns the part-of-speech tags. The lemmatizer and the tagger use a rich feature set that was optimized for all languages of the CoNLL 2009 shared task (Hajič et al., 2009). Our lemmatizer uses the shortest edit script (SES) between the lemmas and the forms and we select a script within an SES list using a MIRA classifier (Chrupala, 2006). The English lemmatizer has an accuracy of 99.46. This is 0.27 percentage point lower than the predicted lemmas of the English corpus in CoNLL 2009, which had an accuracy of 99.73. The German lemmatizer has an accuracy of 98.28. The accuracy of the predicted lemmas in the German corpus was 68.48. The value is different because some closed-class words are annotated differently (Burchardt et al., 2006). We also employed MIRA to train the POS classifiers. Compared to the predicted POS tags in the shared task, we could increase the accuracy by 0.15 from 97.48 to 97.63 for English and by 1.55 from 95.68 to 97.23 for German.

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

4 Dependency Parsing

The dependency parser of this demonstration is a further development of Carreras (2007) and Johansson and Nugues (2008). We adapted it to account for the multilingual corpus of the CoNLL 2009 shared task – seven languages – and to improve the speed of the computationally expensive higher order decoder (Bohnet, 2009). The parser



Parsing sentence required 115ms.

ID	Form	Lemma	PLemma	POS	PPOS	Feats	PFeats	Head	PHead	Deprel	PDeprel	IsPred	Pred	Args: call.03	Args: degree.01
1	Speculators	speculator	speculator	NNS	NNS	_	_	2	2	SBJ	SBJ	_	_	A0	_
2	are	be	be	VBP	VBP	_	_	0	0	ROOT	ROOT	_	_	_	_
3	calling	call	call	VBG	VBG	_	_	2	2	OPRD	OPRD	Y	call.03	_	_
4	for	for	for	IN	IN	_	_	3	3	ADV	ADV	_	_	A1	_
5	a	a	a	DT	DT	_	_	6	6	NMOD	NMOD	_	_	_	_
6	degree	degree	degree	NN	NN	_	_	4	4	PMOD	PMOD	Y	degree.01	_	_
7	of	of	of	IN	IN	_	_	6	6	NMOD	NMOD	_	_	_	A1
8	liquidity	liquidity	liquidity	NN	NN	_	_	7	7	PMOD	PMOD	_	_	_	_
9	that	that	that	WDT	WDT	_	_	10	10	SBJ	SBJ	_	_	_	_
10	is	be	be	VBZ	VBZ	_	_	6	6	NMOD	NMOD	_	_	_	_
11	not	not	not	RB	RB	_	_	10	10	ADV	ADV	_	_	_	_
12	there	there	there	RB	RB	_	_	10	10	LOC-PRD	LOC-PRD	_	_	_	_
13	in	in	in	IN	IN	_	_	12	12	LOC	LOC	_	_	_	_
14	the	the	the	DT	DT	_	_	15	15	NMOD	NMOD	_	_	_	_
15	market	market	market	NN	NN	_	_	13	13	PMOD	PMOD	_	_	_	_
16	.	.	.	.	.	_	_	2	2	P	P	_	_	_	_

Figure 3: The output window. The predicates and their arguments are shown in the upper part of the figure, respectively *call.03* with *A0* and *A1* and *degree.01* with *A1*, while the results in the CoNLL 2008 format are shown in the lower part.

reached the best accuracies in CoNLL 2009 for English and German, and was ranked second in average over all the languages in the task.

The parser in this demonstration is an enhancement of the CoNLL 2009 version with a *hash kernel*, a parallel parsing algorithm, and a parallel feature extraction to improve the accuracy and parsing speed. The hash kernel enables the parser to reach a higher accuracy. The introduction of this kernel entails a modification of MIRA, which is simple to carry out: We replaced the feature-index mapping that mapped the features to indices of the weight vector by a random function. Usually, the feature-index mapping in a support vector machine has two tasks: It maps the features to an index in the weight vector and filters out the features not collected in the first step. The parser is about 12 times faster than a baseline parser without hash kernel and without parallel algorithms. The parsing time is about 0.077 seconds per sentence in average for the English test set.

5 Semantic Role Labeling Pipeline

The pipeline of classifiers used in the semantic role labeling consists of four steps: predicate identification, predicate disambiguation, argument identification, and argument classification, see Fig. 1. In each step, we used different classifiers for the nouns and the verbs. We build all the classifiers using the L2-regularized linear logistic regression from the LIBLINEAR package (Fan et al., 2008). To speed up processing, we disabled the reranker used in the CoNLL 2009 system (Björkelund et al., 2009).

Predicate Identification is carried out using a binary classifier that determines whether a noun or verb is a predicate or not.

Predicate Disambiguation is carried out for all the predicates that had multiple senses in the training corpus. We trained one classifier per lemma. For lemmas that could be both a verb or a noun (e.g. *plan*), we trained one classifier per part of speech. We considered lem-

mas with a unique observed sense as unambiguous.

Argument Identification and Classification. Similarly to the two previous steps, a binary classifier first identifies the arguments and then a multiclass classifier assigns them a label. In both steps, we used separate models for the nouns and the verbs.

Features. For the predicate identification, we used the features suggested by Johansson and Nugues (2008). For the other modules of the pipeline, we used the features outlined in Björkelund et al. (2009). The feature sets were originally selected using a greedy forward procedure. We first built a set of single features and, to improve the separability of our linear classifiers, we paired features to build bigrams.

6 Results and Discussion

The demonstration system implements a complete semantic analysis pipeline for English, where we combined two top-ranked systems for syntactic and semantic dependency parsing of the CoNLL 2009 shared task. We trained the classifiers on the same data sets and we obtained a final semantic F1 score of 80.90 for the full system. This score is lower than the best scores reported in CoNLL 2009. It is not comparable, however, as the predicates had then been manually marked up. Our system includes a predicate identification stage to carry out a fully automatic analysis. This explains a part of the performance drop. To provide comparable figures, we replaced the predicate identification classifier with an oracle reading the gold standard. We reached then a score of 85.58. To reach a higher speed and provide an instantaneous response to the user (less than 1 sec.), we also removed the global reranker from the pipeline which accounts for an additional loss of about 1.2 percentage point. This would put the upper-bound semantic F1 value to about 86.80, which would match the CoNLL 2009 top figures.

Acknowledgments. The research leading to these results has received funding from the European community's seventh framework program FP7/2007-2013, challenge 2, cognitive systems, interaction, robotics, under grant agreement No 230902—ROSETTA.

References

Björkelund, Anders, Love Hafdell, and Pierre Nugues. 2009. Multilingual semantic role labeling. In *Proceedings of CoNLL-2009*.

Bohnet, Bernd. 2009. Efficient parsing of syntactic and semantic dependency structures. In *Proceedings of CoNLL-09*.

Burchardt, Aljoscha, Katrin Erk, Anette Frank, Andrea Kowalski, Sebastian Padó, and Manfred Pinkal. 2006. The SALSA corpus: a German corpus resource for lexical semantics. In *Proceedings of the 5th LREC-2006*.

Carreras, Xavier. 2007. Experiments with a higher-order projective dependency parser. In *Proceedings of CoNLL-2007*.

Chrupala, Grzegorz. 2006. Simple data-driven context-sensitive lemmatization. In *Proceedings of SEPLN*.

Fan, Rong-En, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874.

Hajič, Jan, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The CoNLL-2009 shared task: Syntactic and semantic dependencies in multiple languages. In *Proceedings of CoNLL-2009*.

Johansson, Richard and Pierre Nugues. 2008. Dependency-based syntactic–semantic analysis with PropBank and NomBank. In *Proceedings CoNLL-2008*.

McDonald, Ryan, Koby Crammer, and Fernando Pereira. 2005. Online large-margin training of dependency parsers. In *Proceedings of ACL-2005*.

Shi, Qinfeng, JamesPetterson, Gideon Dror, John Langford, Alex Smola, and S.V.N. Vishwanathan. 2009. Hash kernels for structured data. *Journal of Machine Learning*, 15(1):143–172.

Surdeanu, Mihai, Richard Johansson, Adam Meyers, Lluís Màrquez, and Joakim Nivre. 2008. The CoNLL–2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of CoNLL–2008*.

PanLex and LEXTRACT: Translating all Words of all Languages of the World

Timothy Baldwin,♠ Jonathan Pool♡ and Susan M. Colowick♡

♠ CSSE
University of Melbourne
tb@ldwin.net

♡ Utilika Foundation
{pool,smc}@utilika.org

Abstract

PANLEX is a lemmatic translation resource which combines a large number of translation dictionaries and other translingual lexical resources. It currently covers 1353 language varieties and 12M expressions, but aims to cover all languages and up to 350M expressions. This paper describes the resource and current applications of it, as well as LEXTRACT, a new effort to expand the coverage of PANLEX via semi-automatic dictionary scraping.

1 Introduction

Translation dictionaries, multilingual thesauri, and other translingual lexical (more precisely, lemmatic) resources answer queries of the form "Given lemma X in language A, what possible translations of it into language B exist?" However, existing resources answer only a small fraction of the potential queries of this form. For example, one may find attested translations of the Santiago del Estero Quichua word *unta* into German, English, Spanish, Italian, French, Danish, Aymara, and several other Quechua languages, but not into the other (roughly 7 thousand) languages in the world.

Answers to the vast majority of possible lemmatic translation queries must be inferred. If *unta* can be translated into Spanish as *lleno*, and *lleno* can be translated into Hungarian as *tele*, e.g., perhaps Quichua *unta* can be translated into Hungarian as *tele*. But such inference is nontrivial, because lexical ambiguity degenerates the quality of indirect translations as the paths through intermediate languages grow longer.

	Current	Goal
Resources	766	10K
Language varieties	1353	7000
Expressions	12M	350M
Expression–meaning pairs	27M	1000M
Expression–expression pairs	91M	1000M

Table 1: Current and goal PANLEX coverage

Thus, it appears that the quality and range of lemmatic translation would be supported by an easily accessible graph combining a large (or, ideally, complete) set of translations reported by the world's lexical resources. PANLEX (http://panlex.org) is a project developing a publicly accessible graph of attested lemmatic translations among all languages. As of 2010, it provides about 90 million undirected pairwise translations among about 12 million lemmata in over 1,300 language varieties, based on the consultation of over 750 resources, as detailed in Table 1. By 2011 it is expected that the resources consulted will approximately quadruple.

2 The PANLEX Project

PanLex is an attempt to generate as complete as possible a translation graph, made up of expression nodes, meaning nodes, and undirected edges, each of which links an expression node with a meaning node. Each expression is uniquely defined by a character string and a language. An expression e_i is a translation or synonym of an expression e_j iff there is at least one meaning m_k such that edges $v(e_i, m_k)$ and $v(e_j, m_k)$ exist. For example, *frame* in English shares a meaning with *bikar* in Bahasa Malay, and *bikar* shares a meaning with *beaker* in English, but *frame* shares no

Coling 2010: Demonstration Volume, pages 37–40,
Beijing, August 2010

meaning with *beaker*. Whether e_i and e_j are synonyms or translations depends on whether their languages are identical. In Table 1, "expression–meaning pairs" refers to edges $v(e,m)$ and "expression–expression pairs" refers to expressions with at least one meaning in common.

2.1 Current Applications of PANLEX

While lemmatic translation falls short of sentential and discourse translation, it is not without practical applications. It is particularly useful in author–machine collaborative translation, when authors are in a position to lemmatize expressions. The prototype PANIMAGES application (`http://www.panimages.org`), based on PANDICTIONARY, elicits a lemmatic search query from the user and expands the query into dozens of languages for submission to image-search services. Hundreds of thousands of visitors have used it to discover relevant images labeled in languages they do not know, sometimes selecting particular target languages for cultural specificity or to craft less ambiguous queries than their own language would permit (Christensen et al., 2009).

In lemmatic messaging applications developed for user studies, users lemmatized sentences to tell stories or send mail across language boundaries. Even with context-unaware translation of lemmata producing mostly non-optimal translations, users were generally able to reconstruct half or more of the originally intended sentences (Soderland et al., 2009). The PanLex database was also used in a multilingual extension of the image-labeling game initiated by Von Ahn and Dabbish (2004).

User and programmatic interfaces to PanLex are under development. A lemmatic user interface (`http://panlex.org/u`) communicates with the user in a potentially unlimited set of languages, with PanLex dynamically using its own data for the localization. A primitive API makes it possible for developers to provide, or make infrastructural use of, lemmatic translation via PanLex. Prototype lemmatic translation services like TeraDict (`http://panlex.org/demo/treng.html`), InterVorto (`http://panlex.org/demo/trepo.html`), and TümSöz (`http://panlex.org/demo/trtur.html`) exploit the API.

2.2 Extraction and Normalization

The approach taken by PANLEX to populate the translation graph with nodes and edges is a combination of: (a) extraction of translation pairs from as many translingual lexical resources as can be found on the web and elsewhere; and (b) inference of new edges between expressions that exist in PANLEX.

To date, extraction has taken the form of hand writing a series of regular expression-based scripts for each individual dictionary, to generate normalized PANLEX database records. While this is efficient for families of resources which adhere to a well-defined format (e.g. FREEDICT or STARDICT dictionaries), it does not scale to the long tail of one-off dictionaries constructed by lexicographers using ad hoc formats, as detailed in Section 2.2. LEXTRACT is an attempt to semi-automate this process, as detailed in Section 3.

Inference of new translation edges is nontrivial, because lexical ambiguity degenerates the quality of indirect translations as the paths through intermediate languages grow longer. PANDICTIONARY is an attempt to infer a denser translation graph from PANLEX combining translations from many resources based on path redundancy, evidence of ambiguity, and other information (Sammer and Soderland, 2007; Mausam et al., 2009; Mausam et al., 2010).

PANLEX is more than a collection, or docbase, of independent resources. Its value in translation inference depends on its ability to combine facts attested by multiple resources into a single graph, in which lemmata from multiple resources that are substantively identical are recognized as identical. The obstacles to such integration of heterogeneous lexical data are substantial. They include: (1) ad hoc formatting, including format changes between portions of a resource; (2) erratic spacing, punctuation, capitalization, and line wrapping; (3) undocumented and non-standard character encodings; (4) vagueness of the distinction between lemmatic (e.g. *Rana erythraea*) and explanatory translations (e.g. *a kind of tree frog*); and (5) absence of consensus for some languages as to the representation of lemmata, e.g. hyphenation and prefixation in Bantu languages, and inclusion or exclusion of tones in tonal languages.

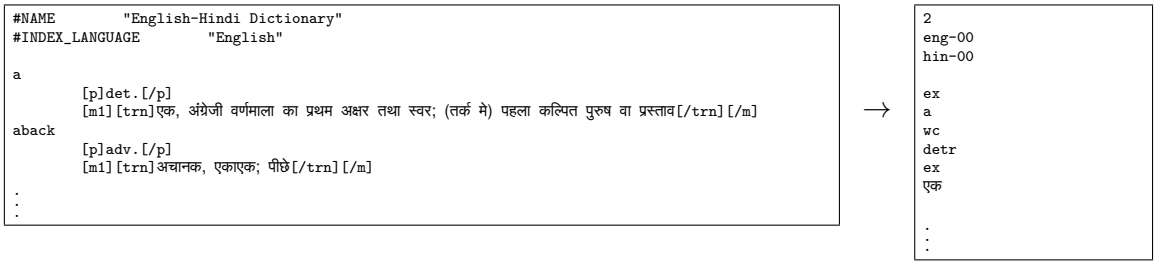


Figure 1: A snippet of an English–Hindi dictionary, in its source form (left) and as normalized PANLEX records (right)

3 LEXTRACT

LEXTRACT is a sub-project of PANLEX, aimed at automating the extraction and normalization of data from arbitrary lexical resources, focusing in the first instance on text-based resources, but ultimately including XML, (X)HTML, PDF and wiki markup-based resources. The approach taken in LEXTRACT is to emulate the manual workflow used by the PANLEX developers to scrape data from dictionary files, namely learning of series of regular expressions to convert the source dictionary into structured database records. In this, we assume that the source dictionary has been transcoded into utf-8 encoding,[1] and further that the first five PANLEX translation records found in the source dictionary have been hand generated as seed instances to bootstrap the extraction process off, as illustrated in Figure 1. Briefly, this provides vital data including: specification of the source and target languages; manual disambiguation of expression–expression vs. expression–meaning structuring; any optional fields such as part of speech; and (implicitly) where the records start from in the source file, and what fields in the original dictionary should not be preserved in the PANLEX database.

The procedure for learning regular expressions can be broken down into 3 steps: (1) record matching; (2) match lattice pruning; and (3) regular expression generalization.

Record matching involves determining the set of codepoint spans in the original dictionary where the component strings (minimally the source and target language expressions, but possibly including domain information, word class information or other metadata) encoded in the five seed records can be found, to use as the basis for learning the formatting idiom employed in the dictionary. For each record, we determine all positions in the source dictionary file where all component strings can be found within a fixed window width of one another. This is returned as a match lattice, representing the possible sub-extents ("spans") in the source dictionary of each record, and the location(s) of each component string within each.

Match lattice pruning takes the match lattice from the record matching step, and prunes it based on a combination of hard and soft constraints. The single hard constraint currently used at present is that the records must occur in the lexicon in sequence; any matches in the lattice which violate this constraint can be pruned. Soft constraints include: each record should span the same number of lines; the fields in each record should occur in the same linear order; and the width of the inter-field string(s) should be consistent. These are expectations on dictionary formatting, but can be violated (e.g. a given dictionary may have some entries on a single line and others spanning two lines). To avoid over-pruning the lattice, we determine the coverage of each such soft constraint in the form of: (a) *type*-level coverage, i.e. the proportion of records for which a given constraint setting (e.g. record size in terms of the number of lines it spans) matches with at least one record span; and (b) *token*-level coverage, i.e. the proportion of individual spans a given constraint setting matches. We apply soft constraints conservatively, selecting the soft constraint setting with full type-level coverage (i.e. it matches all records)

[1] We have experimented with automatic character encoding detection methods, but the consensus to date has been that methods developed for web documents, such as the CHARDET library, are inaccurate when applied to dictionary files.

and maximum token-level coverage (i.e. it prunes the *least* edges in the lattice). Soft constraints are applied iteratively, as indicated in Algorithm 1.

Algorithm 1 Match lattice pruning algorithm

```
1: Initialize l                          ▷ initialize record matching match lattice
2: repeat
3:     change ← False
4:     for all h_i ∈ H do                ▷ update hard constraint coverage
5:         (htype_i, htoken_i) ← coverage(h_i, l)
6:         if htoken_i < 1 then          ▷ if pruneable edges
7:             l ← apply(h_i, l)         ▷ apply constraint
8:             change ← True
9:         end if
10:    end for
11:    for all s_i ∈ S do                ▷ update soft constraint coverage
12:        {(stype_ij, stoken_ij)} ← coverage(c_i, l)
13:    end for
14:    if s ← arg max_{s_ij}(∃stype_ij = 1.0 ∧ stoken < 1.0 ∧
       (∀i' ≠ i : |stype_i*| > 1, ∀j' : stoken_ij < 1.0 : stoken_ij >
       stoken_i'j')) then
15:        l ← apply(s, l)               ▷ apply constraint
16:        change ← True
17:    end if
18: until change = False
```

The final step is regular expression generalization, whereby the disambiguated match lattice is used to identify the multiline span of all records in the source dictionary, and inter-field strings not corresponding to any record field are generalized across records to form a regular expression, which is then applied to the remainder of the dictionary to extract out normalized PANLEX records. As part of this, we build in dictionary-specific heuristics, such as the common practice of including optional fields in parentheses.

The LEXTRACT code is available from `http://lextract.googlecode.com`.

LEXTRACT has been developed over 10 sample dictionaries, and record matching and match lattice pruning has been found to perform with 100% precision and recall over the seed records. We are in the process of carrying out extensive evaluation of the regular expression generalization over full dictionary files.

Future plans for LEXTRACT to get closer to true emulation of the manual extraction process include: dynamic normalization of target language strings (e.g. normalizing capitalization or correcting inconsistent pluralization) using a combination of language-specific tools for high-density target languages such as English, and analysis of existing PANLEX expressions in that language; elicitation of user feedback for extents of the document where extraction has failed, fields where the correct normalization strategy is unclear (e.g. normalization of POS tags not seen in the seed records, as for *det.* →*detr* in Figure 1); and extending LEXTRACT to handle (X)HTML and other file types.

References

Christensen, Janara, Mausam, and Oren Etzioni. 2009. A rose is a roos is a ruusu: Querying translations for web image search. In *Proc. of the Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2009)*, pages 193–196, Suntec, Singapore.

Mausam, Stephen Soderland, Oren Etzioni, Daniel Weld, Michael Skinner, and Jeff Bilmes. 2009. Compiling a massive, multilingual dictionary via probabilistic inference. In *Proc. of the Joint conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing (ACL-IJCNLP 2009)*, pages 262–270, Suntec, Singapore.

Mausam, Stephen Soderland, Oren Etzioni, Daniel S. Weld, Kobi Reiter, Michael Skinner, Marcus Sammer, and Jeff Bilmes. 2010. Panlingual lexical translation via probabilistic inference. *Artificial Intelligence*, 174(9–10):619–637.

Sammer, Marcus and Stephen Soderland. 2007. Building a sense-distinguished multilingual lexicon from monolingual corpora and bilingual lexicons. In *Proc. of the Eleventh Machine Translation Summit (MT Summit XI)*, pages 399–406, Copenhagen, Denmark.

Soderland, Stephen, Christopher Lim, Mausam, Bo Qin, Oren Etzioni, and Jonathan Pool. 2009. Lemmatic machine translation. In *Proc. of Machine Translation Summit XII*, page 2009, Ottawa, Canada.

Von Ahn, Luis and Laura Dabbish. 2004. Labeling images with a computer game. In *Proc. of the SIGCHI conference on Human factors in computing systems*, pages 319–326, Vienna, Austria.

Antelogue: Pronoun Resolution for Text and Dialogue

Eleni Miltsakaki
University of Pennsylvania
elenimi@seas.upenn.edu

Abstract

Antelogue is a pronoun resolution prototype designed to be released as off-the-shelf software to be used autonomously or integrated with larger anaphora resolution or other NLP systems. It has modules to handle pronouns in both text and dialogue. In *Antelogue*, the problem of pronoun resolution is addressed as a two-step process: a) acquiring information about properties of words and the entities they represent and b) determining an algorithm that utilizes these features to make resolution decisions. A hybrid approach is implemented that combines known statistical and machine learning techniques for feature acquisition and a symbolic algorithm for resolution.

1 Introduction

Pronoun resolution is the well-known problem of identifying antecedents for pronominal references in text or dialogue. We present a prototype of new system for pronoun resolution, *Antelogue*, that handles both text and dialogues. In our approach, pronoun resolution is done in two steps: a) feature acquisition of properties of words and the entities they represent and b) resolution algorithm. We adopt a hybrid approach to the problem, using statistical and machine learning techniques widely available in the NLP literature to collect features and a symbolic algorithm informed by prior research in anaphora resolution and models of entity salience to appropriately rank and evaluate antecedents.

The design and architecture of *Antelogue* is modular and flexible and will soon be released for off-the-shelf use as an independent component or for possible integration of larger anaphora resolution systems, such as the GuiTAR (General Tool for Anaphora Resolution) (Poesio and Kabadjov, 2004) that currently is released with (Mitkov et al., 2002)'s statistical pronoun resolution algorithm, MARS, that processes pronouns in text. Motivation for building a new algorithm for text and dialogues has been the problem of alignment between caption dialogues and stage directions on one hand and video content in movies on the other. While pronoun resolution in stage directions proved to be a fairly easy task, in dialogues we are facing the following challenges:

1. Part of speech taggers trained on text (typically the Wall Street Journal texts of Penn Treebank) perform poorly on dialogues, primarily due to the fragmented nature of spoken language. As a result NP tags are overgenerated.
2. Fragmentary speech and disfluencies or false starts common in dialogues cannot be handled by parsers trained on text.
3. First and second person pronouns are common. Special algorithms are needed to handle them.
4. Special *addressee* patterns need to be identified to block first and second person named references (e.g., "Hey, John, where did he go?") becoming antecedents for third person pronouns.
5. In dialogues, pronouns can be used for reference to people or objects that are visually but not textually accessible. Special algorithms are needed to identify when an antecedent is not present in the text.
6. Pronouns are used for reference to people or objects that are visually salient in the scene but not mentioned explicitly in the dialogue, i.e., there are no textual antecedents.
7. Multi-party dialogues, sometimes 3rd person pronouns are used to refer to other speakers. It is hard to identify when an instance of a 3rd person pronoun has an antecedent in the prior discourse

Coling 2010: Demonstration Volume, pages 41–44,
Beijing, August 2010

or another speaker.

In what follows, we present the system's design and architecture and the components that have already been implemented. In the demo, the users will be able to use *Antelogue*'s GUI to enter their own data and evaluate the system's performance in real time. The current version handles first, second, and third person singular pronouns, including a classification recognizing referential and non-referential instances of "it". *Antelogue* does not, yet, handle plural pronouns or recognize impersonal uses of singular "you".

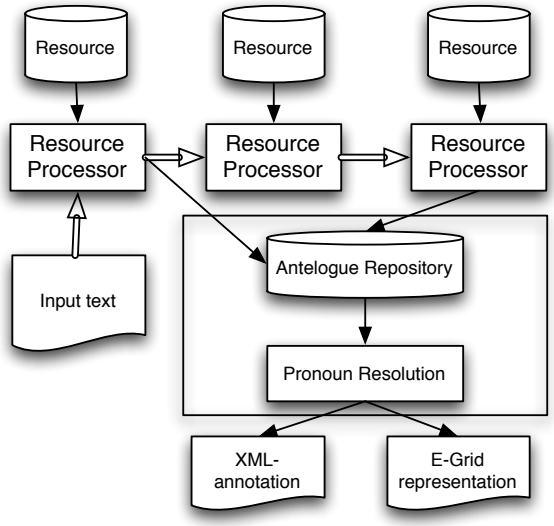


Figure 1: General System Architecture

2 System design

The problem of pronoun resolution is addressed as a two-step process: a) acquiring information about properties of words and the entities they represent and b) determining an algorithm that utilizes these features to make resolution decisions. A hybrid approach is implemented that combines known statistical and machine learning techniques for feature acquisition and a symbolic algorithm for resolution.

For the feature acquisition step, any number of feature acquisition sub-modules can be implemented. The architecture is flexible such that new feature acquisition modules can be added as they may become available or deemed crucial for specific applications. The demo version acquires features from a sentence tokenizer, word tokenizer, NER tagger, gender and number database and POS tagger. For every sub-module a corresponding parser analyzes the output of the submodules to retrieve the features and store them in the *Antelogue repository*.

The resolution step implements an algorithm for utilizing the features in the repository to make resolution decisions. The resolution module needs to communicate only with the repository to get feature information and outputs xml annotated text or, what we call, *e-grid* output in which pronouns have been replaced by their antecedents. If the identified antecedent is a pronoun, it is further looked-up until a non-pronominal antecedent is found. A pronominal antecedent is shown only in case there is no nominal antecedent available.

The architecture of Antelogue is illustrated in Fig. 1. *Antelogue* can be set to perform pronoun resolution in both dialogue and text. A preprocessing step is required to ensure that the files are in the appropriate format. Because *Antelogue* was built to perform pronoun resolution in the dialogues and stage directions of screenplays, the pre-processing steps required to extract dialogues and text from the TV series*Lost*, are available.

3 System architecture

Feature acquisition *Sentence and word tokenization:* built based on (Ratnaparkhi, 1996). To address dialogue idiosyncrasies, sentence tokenization is forced to respect speaker turns thus blocking forming sentences across speaker turns.
Word processor. This module processes the word tokenized file and creates an indexed entry for every word in the *Antelogue repository*.
Named Entity Recognizer tagging (NER): We integrated Stanford's NER tagger (Finkel et al., 2005).
NER processor. This module processor the NER tagged file and associates identified NER tags with the corresponding words in the *Antelogue repository*.
Gender and Animacy processor: This modules collects gender information from the gender corpus[1] (Bergsma and Lin, 2006) and checks a self-

[1] http://www.cs.ualberta.ca/~bergsma/Gender.

made corpus for profession (teacher, doctor, etc) and family relations (mother, father, etc), extracted from web searches. In the gender corpus, gender and number data are collected statistically and are not always reliable. We developed a simple confidence metric to evaluate the reliability of the gender and number data. If the ratio of the highest probability to the sum of all other probabilities is lower than 60% we mark gender or number unknown.[2] *Part-of-speech tagging (POS)*. We trained (Ratnaparkhi, 1996)'s POS tagger on dialogue data obtained from the English CTS Treebank with Structural Metadata released by LDC in 2009. *POS parser*. This modules parses the POS-tagged input and updates the *Antelogue repository*.

Pronoun resolution The pronoun resolution submodule, currently, has three submodules: a) first and second person pronouns, b) third person singular masculine and feminine pronouns, and c) third person singular neuter pronouns.

For the first and second person pronouns, *Antelogue* identifies and resolves all instances of "I" to the speaker name and all instances of "you" to the next speaker. It there is no other speaker (when "you" is in the last turn), the algorithm will pick the speaker from the previous turn. If there is no previous turn, it is declared unresolvable.

For the third person "he" and "she" module, the algorithm*Antelogue* searches for pronouns backwards starting at the last sentence of the dialogue. For every sentence we construct a list of potential antecedents identified as nouns or pronouns by the POS tagger. A number of filters, then apply, to filter out incompatible antecedents. A category of incompatible antecedents for 'he' and 'she' that is almost unique to dialogues are addressee references. We identify references to addressee using surface punctuation features. Resolution starts with a look-up at antecedents of the current sentences, processing them from left-to-right. If the first antecedent is identified in the human corpus and has compatible gender information, it is picked. If not, the gender corpus is searched for reliable matches. Once a match is identified, it is filtered by NER. The gender corpus often assigns feminine or masculine gender to common nouns. Only those entities that have a NER tag pass the compatibility test. If no compatible antecedent is found in the current sentence, *Antelogue* continues search in the previous sentence. If the dialogues have scene boundaries, as the case is in *Lost*, the search for an antecedents stops at a scene boundary. Otherwise it will not stop before the first sentence of the dialogue is reached. If no compatible antecedent is found, it is declared 'unresolvable'. Correctly declaring pronouns unresolvable is extremely useful in dialogues, especially from movies, in which a referent of a third person pronoun may be visually available but not introduced in the prior discourse. Correctly unresolvable feminine and masculine pronouns signal a cue for search in the visuals scene, a cross-modal direction that we are pursuing as part of future work.

For the third person "it", we first need to address the issue of identifying referential and non-referential instances of "it".[3] Non-referential instances of "it" include pleonastic "it" (e.g., "it rains", or "it is certain that...") and references to a verbal or other clausal antecedent (e.g., "it" in "Mary got the award. It's wonderful!). For the "it" classification task, we follow (Bergsma et al., 2008)' approach. We generate 4 and 5 word patterns out using the found occurrences of "it' then replace "it/its" with "they/theirs/them". Frequencies of the substituted versions are computed using data from the Google n-gram corpus. If substitutions with "they/theirs/them" are not common, "it" is classified as non-referential.

Antelogue outputs a)an XML file with annotations of entities, pronouns and antecedents, and b)an "e-grid representation file" in which all pronouns have been replaced with their referents. In the XML file, pronouns are either resolved or declared unresolvable if no antecedent is identified. The pronoun "it" can, additionally, be declared non-referential. The e-grid representation file is useful for evaluating text coherence using the file directly as input to the (Barzilay and Lapata, 2008)'s e-grid model, a direction we want

[2](Charniak and Elsner, 2009)'s system 'learns' gender information using Expectation Maximization.

[3]For simplicity, we are sloppy here using the term non-referential to mean non-referring to a nominal entity.

to take in the future to explore its strengths in automatically identifying scene boundaries. Despite well-known problems in making meaningful comparisons in pronoun resolution systems, *Antelogue*'s performance is comparable to some of the highest reported performances, either identifying correctly an antecedent or correctly declaring a pronoun unresolvable or non-referential in 85% of 600 annotated pronouns.

Text module: *Antelogue*'s architecture for resolving pronouns in text is identical to dialogues except that a)the pre-processing text extracts text from the stage directions in the screenplay, b) addressee patterns are not used to filter out antecedents for "he" and "she" and instances of "I" and "you" are ignored. In the future we plan to implement resolution of "I" and "you" as well as a dialogue style resolution of "he" and "she" for instances of embedded speech. These instances were extremely rare in our data but they need to be catered for in the future. *Antelogue*'s performance exceeds 90% for stage directions because stage directions are relatively simple and fairly unambiguous. For this reason, a syntactic parse which slows down the system considerably was not used. However, to retain similar levels of performance in different domains, the use of syntactic parse will be needed.

4 Antelogue API and demo

Antelogue is implemented in Java. Its API includes an executable file, an empty database for the repository and command line instructions for running the system. The dialogue POS tagger is also available. The other feature acquisition submodules, text POS tagger, NER tagger and gender database are publicly available. Antelogue makes use of the google n-gram corpus, available through the Linguistic Data Consortium.[4]

As an off-the-shelf application, designed both for integration but also for experimentation, evaluation and comparison with other systems, *Antelogue* runs on a single unix command. The user is prompted to choose the dialogue or text module and then is asked to determine the path with the data. *Antelogue* returns annotated files with resolved pronouns in seconds for a reasonably sized file (approx. 2,000-3,000 words) or in couple of minutes for very large files. These processing time estimates apply to the demo version. Processing time will carry depending on the number of submodule implemented in the feature acquisition step.

For the demo, we built a special Graphical User Interface. In the left part of the GUI, the user can either type in his or her own text or dialogue, paste text or dialogue, or select a local file. There are selections for the text/dialogue mode and xml/e-grid outputs. *Antelogue* performs pronoun resolution in real time and show the results on the right hand side part of the GUI.

Acknowledgments: Special thanks to Ben Taskar for his help and guidance in this project and to NSF IIS-0803538 grant for financial support.

[4] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2006T13

References

Barzilay, R. and M. Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*.

Bergsma, S. and D. Lin. 2006. Bootstrapping path-based pronoun resolution. In *ACL'06*, pages 33–40.

Bergsma, S., D. Lin, and R. Goebel. 2008. Distributional identification of non-referential pronouns. In *ACL'08*, pages 10–18.

Charniak, E. and M. Elsner. 2009. Em works for pronoun resolution. In *Proceedings of EACL 2009*.

Finkel, J.R., T. Grenager, and C. Manning. 2005. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Ann Arbor*, 100.

Mitkov, R., R. Evans, and C. Orasan. 2002. A new, fully automatic version of Mitkov's knowledge-poor pronoun resolution method. *Lecture notes in computer science*, pages 168–186.

Poesio, M. and M.A. Kabadjov. 2004. A general-purpose, off-the-shelf anaphora resolution module: Implementation and preliminary evaluation. In *Proc. of the 4th International Conference on Language Resources and Evaluation. Lisbon, Portugal*. Citeseer.

Ratnaparkhi, A. 1996. A maximum entropy model for part-of-speech tagging. In *In Proceedings of EMNLP'96*, pages 133–142.

E-HowNet and Automatic Construction of a Lexical Ontology

Wei-Te Chen, Su-Chu Lin, Shu-Ling Huang, You-Shan Chung, and **Keh-Jiann Chen**
Institute of Information Science, Academia Sinica
weitehchen@gmail.com,
{jess, yosieh, yschung, kchen}@iis.sinica.edu.tw

Abstract

In this paper, we propose a lexical senses representation system called E-HowNet, in which the lexical senses are defined by basic concepts. As a result, the meanings of expressions are more specific than those derived by using primitives. We also design an ontology to express the taxonomic relations between concepts and the attributes of concepts. To establish the taxonomic relations between word senses, we introduce a strategy that constructs the E-HowNet ontology automatically. We then implement the lexical ontology as a Web application[1] to demonstrate the taxonomy and the search functions for querying key-terms and E-HowNet expressions in the lexicon, which contains more than 88,000 lexical senses.

1 Introduction

E-HowNet, an evolution and extension of HowNet (Dong & Dong, 2006), is an entity-relation representation model for lexical senses. Under the framework, word senses are defined by basic concepts as well as conceptual relations called attribute-values. The following is an example of lexical sense representation in E-HowNet.

(1) '慎 選|carefully choose' is expressed (or defined) by the expression '{choose|選擇:manner={cautious|慎}}'.

In the representation, the meaning of "慎選" is comprised of two primitive concepts, "choose|選擇" and "cautious|慎", and the conceptual relation between the primitives is explained by the semantic role "manner". For further details, readers may refer to the E-HowNet technical report (CKIP 2009).

With a well-established entity-relation model, semantic composition is applicable from the morphological level to the sentential level in E-HowNet. Semantic compositionality, together with syntactic information, contributes enormously to natural language understanding.

The remainder of this paper is organized as follows. We describe the major features of E-HowNet in Section 2 and introduce the E-HowNet ontology in Section 3. Then, we present our online E-HowNet system in Section 4. Section 5 contains some concluding remarks.

To achieve the goal of semantic compositionality and to extend the advantage from HowNet, the following features are implemented in E-HowNet.

a) Multi-level definitions and semantic decomposition: Word senses (concepts) can be defined (expressed) by primitives as well as by any well-defined concepts and conceptual relations. However, using only primitives to express word senses, as in HowNet, causes information degradation and important ontological relations between concepts may be missed.

b) Uniform sense representation and semantic compositionality: To achieve semantic compositionality, it is necessary to encode the senses of both content words and function words in a uniform framework. HowNet performs well for defining content words, but it does not provide a well-form representational framework for expression the sense of function words, which indicate semantic relations. In contrast, E-HowNet

[1] available at http://ckip.iis.sinica.edu.tw/~wtchen/taxonomy/

Coling 2010: Demonstration Volume, pages 45–48,
Beijing, August 2010

provides uniform representations for the senses of content/function words and the senses of sentences/phrases. For example, the passive sense of the preposition '被 by' introduces an agent role (relation) and the conjunction '因為 because' links the relation of reason between two events. The functional representation and semantic compositionality are illustrated by the following example:

(2) Because of the rain, the clothes are all wet.
因為下雨，衣服都濕了。

Table 1: The function representation and semantic compositionality for example sentence

Word	POS	E-HowNet Definition
因為	Cb (conjunction)	reason ={ }
下雨	VA (intransitive verb)	{rain|下雨}
衣服	Na (common noun)	{clothing|衣物}
都	Da (adverb)	Quantity= {complete|整}
濕	VH (state verb)	{wet|濕}
了	Ta (particle)	aspect= {Vachieve|達成}

Suppose that the following dependency structure and semantic relations are derived by parsing sentence (2) as follows:

(3) S(reason:VP(Head:Cb:因為|dummy:VA:下雨)|theme:NP(Head:Na:衣服) | quantity: Da:都 | Head:Vh:濕|particle:Ta:了)。

The semantic composition in (4) is the result of unifying the features of the lexical representations shown in the above table. The dependency daughters have become feature attributes of the sentential head 'wet|濕'.

(4) def:{wet|濕:
theme={clothing|衣物},
aspect={Vachieve|達成},
quantity={complete|整},
reason={rain|下雨}}.

c) Taxonomy for both entities and relations: To achieve automatic feature unification, E-HowNet organizes entities and relations (attributes) in a hierarchical structure that relates entities taxonomically. Further details are provided in the next section.

2 Ontology

We adopt and extend approximately 2,600 primitives from HowNet to form the top-level ontology of E-HowNet, which includes two types of subtrees: entities and relations. The entities are comprised of events, objects, and attribute-values; while the relations are comprised of semantic-roles and functions. Entities indicate concepts that have substantial content, whereas relations link the semantic relations between entities (Chen et al., 2004; Chen et al., 2005; Chen et al., 2005; Huang et al, 2008). The taxonomic structure is organized by hypernym-hyponym relations; therefore, it forms an inheritable system, i.e., the hyponym concepts inherit the properties of hypernym concepts. The proposed approach facilitates the adoption of knowledge represented by other frameworks, such as FrameNet, and HowNet; and it allows concepts to be represented with varying degrees of specificity. Another advantage is that conceptual similarities can be modeled by their relational distances in the hierarchy (Resnik, 1999), and the taxonomic relations between lexical senses can be captured from their E-HowNet expressions automatically.

2.1 Automatic Construction of Ontology

With E-HowNet expressions, lexical senses are defined as entities and relations. Thus, all the taxonomic relations of lexical senses can be identified according to their E-HowNet definitions. Synonyms are identified by their identical E-HowNet expressions, and hyponymy relations are identified by the subsumption of attribute-values. (Note that only near-synonym classes are identified due to the coarse-grained expressions of the lexical senses in the current version of E-HowNet.) Furthermore, new categories are identified by common attribute-values. For instance, pandas and zebras can be categorized as animals with the same feature: black and white markings. To construct a complete lexical taxonomy, we use

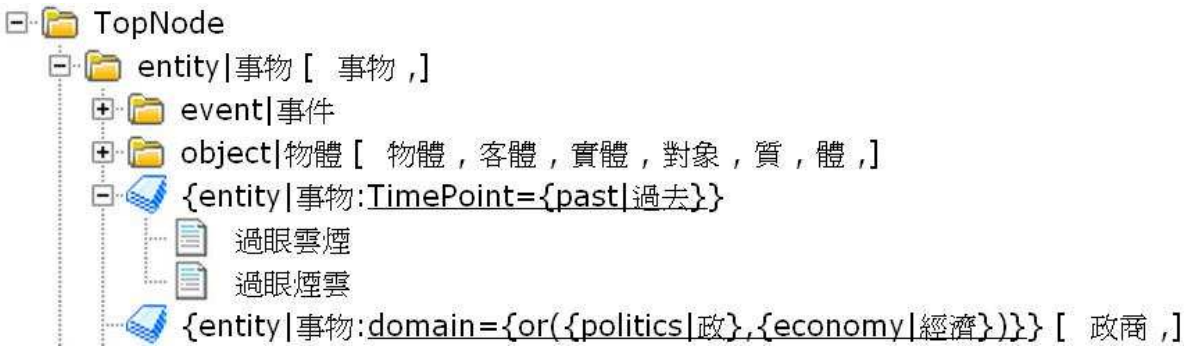


Figure 1: The E-HowNet ontology system

a strategy that categorizes concepts automatically.

Starting with a manually created top-level ontology of primitive concepts, the following strategy classifies the lexicon into hierarchical sub-categories:

(1) Attach lexical senses. Words and associated sense expressions are first attached to the top-level ontology nodes according to their head concepts. For instance, the head concept of the expression '{choose|選擇:manner={cautious|慎}}' is 'choose|選擇'.

(2) Sub-categorization by attribute-values. Lexical concepts with the same semantic head are further sub-categorized according to their attribute-values. Lexicons that have the same attribute-values share specific characteristics; therefore further sub-categorization is performed based on the distinct attribute-values of the lexicons.

(3) Repeat step (2) if there are too many lexical concepts in one category. Although the lexicons are classified after step (2), some sub-categories might still contain too many lexicons. In this situation, we further classify the lexicons in the sub-category with other attribute-values until all sub-categories contain fewer members than a predefined threshold, or all members of a category are synonyms.

3 Overview of the On-line System

The current E-HowNet ontology is an on-line version of the automatically constructed taxonomic structure of E-HowNet expressions, which contain more than 88,000 lexical senses. This section provides an overview of the ontology and the functions of the on-line web browsing system.

Figure 2: Key-Term Search Box

Figure 1 shows the E-HowNet ontology system and tree structure.

The tree structure of hyponymy relations allows users to browse the entire tree by expanding and hiding sub-trees. Although the classification strategy enables the number of entities under each node to be limited and viewed easily, a more effective function is essential for exploring more than 88 thousand items of data in E-HowNet. Therefore, we provide a search function that allows users to query lexical senses in two ways:

Key-Term Search: The first way is key-term search, which is shown in Figure 2. The syntax of the query interface is like that used by conventional search engines. By inputting the key-term "物體" , the system will search all the taxonomy nodes, sub-categories, and lexical nodes. Then, the results for the taxonomy node "object|物體" and the lexical word "物體" will be displayed in

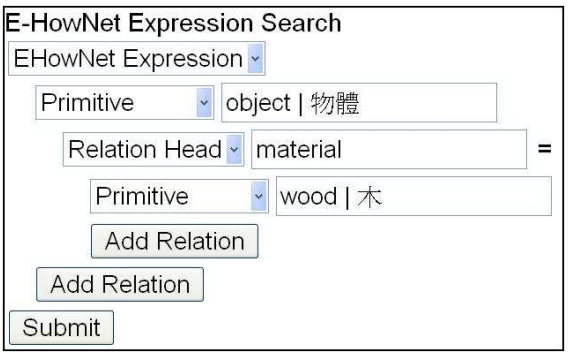


Figure 3: E-HowNet Expression Search Box

the respective columns.

E-HowNet Expression Search: To search a class of words with specific attribute-values, we provide another query syntax for exploring data in E-HowNet Expression. For instance, to find all expressions about wooden objects involves finding E-HowNet data items containing the entity "object—物體" and the attribute-value "material={wood|木}". The expressions are entered on the form shown in Figure 3 and submitted to the system. The results of word senses denoting wooden objects are then returned.

4 Conclusion

E-HowNet sense representations are incremental. Hence, lexical sense expressions can be updated and refined at anytime. In addition, logical relations and the taxonomic structure can be rebuilt automatically based on the refined expressions. New categories in the taxonomy can be identified and characterized by their specific attribute-values. Uniform representations of function words and content words facilitate semantic composition and decomposition, and allow users to derive sense representations of phrases/sentences from the composition of lexical senses. Furthermore, because of E-HowNet's semantic decomposition capability, the primitive representations for surface sentences with the same deep semantics are nearly canonical. We have implemented the E-HowNet ontology online to demonstrate the taxonomy, sub-categories, and lexicons in a hierarchical tree structure. In addition, we provide search functions for querying key-terms and E-HowNet expressions.

References

Keh-Jiann Chen, Shu-Ling Huang, Yueh-Yin Shih and Yi-Jun Chen. 2004. Multi-level Definitions and Complex Relations in Extended-HowNet. In *Proceedings of the Fifth Workshop on Chinese Lexical Semantics 2004*, Beijing University. (in Chinese)

Keh-Jiann Chen, Shu-Ling Huang and Yueh-Yin Shih, Yi-Jun Chen. 2005. Extended-HowNet- A Representational Framework for Concepts. In *Proceedings of OntoLex 2005*, Jeju Island, South Korea.

Yi-Jun Chen, Shu-Ling Huang, Yueh-Yin Shih and Keh-Jiann Chen. 2005. Semantic Representation and Definitions for Function Words in Extended-HowNet. In *Proceedings of the Sixth Workshop on Chinese Lexical Semantics 2005*, Xiamen University.

Z. D. Dong and Q. Dong 2006. HowNet and the Computation of Meaning. World Scientific Publishing Co. Pte. Ltd.

Shu-Ling Huang, Shih Yueh-Yin and Keh-Jiann Chen 2008. Knowledge Representation for Comparison Words in Extended-HowNet. *Language and Linguistics*, vol. 9(2), pp. 395-414.

Philip Resnik. 1999. Semantic similarity in a Taxonomy: An information-based measure and its application to problems of ambiguity in natural language. *Journal of Artifical Intelligence Research*, vol. 11, pp. 95-130.

CKIP. 2009. Lexical Semantic Representation and Semantic Composition: An Introduction to E-HowNet (E-HowNet Technical Report). Academia Sinica, Taipei.

Cloud Computing for Linguists

Dorothee Beermann
Norwegian University of Science and Technology
dorothee.beermann@hf.ntnu.no

Pavel Mihaylov
Ontotext
pavel@ontotext.com

Abstract

The system presented is a web application designed to aid linguistic research with data collection and online publishing. It is a service mainly for linguists and language experts working with language description of less-documented and less-resourced languages. When the central concern is in-depth linguistic analysis, maintaining and administering software can be a burden. Cloud computing offers an alternative. At present mainly used for archiving, we extend linguistic web applications to allow creation, search and storage of interlinear annotated texts. By combining a conceptually appealing online glosser with an SQL database and a wiki, we make the online publication of linguistic data an easy task also for non-computationally oriented researchers.

1 General description of TypeCraft

TypeCraft (or TC in short) is a multilingual online database of linguistically-annotated natural language texts, embedded in a collaboration and information tool. It is an online service which allows users (projects as well as individuals) to create, store and retrieve structured data of the kind mainly used in natural language research. In a system featuring graded access the user may create his own domain, invite others, as well as share his data with the public. The kernel of TypeCraft is morphological word level annotation in a relational database setting, wrapped into a wiki which is used as a communication and information gathering and sharing tool. TypeCraft allows the import of raw text for storage and annotation and export of annotated data to MS Word, OpenOffice.org, LaTeX and XML. The online system is complemented by an offline client which is a Java application offering the same functionality as the online version. This allows a seamless exchange of data between the server and the user's own computer.

2 Online system internals

The online system is supported by a central server running the following modules: TypeCraft server proper, an SQL database, Apache, MediaWiki. The client side consists of the TypeCraft editor interface and a wiki environment (content produced by MediaWiki on the server). Users perceive the wiki and the editor interface as a single TypeCraft web application.

The TypeCraft server proper is a Java application running inside a Java application server. TypeCraft uses a PostgreSQL database for data storage. The data mapping between Java objects and database tables is managed by Hibernate, so the system is not bound to any specific SQL database. TypeCraft data can be divided into two distinct groups: *common data*, shared between all annotated tokens and users, such as the word and sentence level tag sets and an ISO 639-3 specification, and *individual data*, by which we mean specific texts, phrases, words and morphemes. Individual data references common data types. This for example means that all users of the system making use of the part of speech tag N share the reference to a single common tag N.

3 Digital linguistic data

It is well known that generation of linguistic annotation of any kind is a time consuming enterprise quite independent of the form the primary data has and the tools chosen for processing this data. Equally well known are problems connected to the generation and storage of linguistic data.

Coling 2010: Demonstration Volume, pages 49–52,
Beijing, August 2010

Standard word processing programs do not function well as linguistic tools and private computers are not a safe place to store linguistic resources (Bird and Simons, 2003). Although it is generally agreed that linguistic resources should be kept in a sustainable and portable format, it is less clear what that really means in practice. For the individual researcher it is not easy to decide which of the available tools serve his purpose best. To start with, it is often unclear which direction the research will take, which categories of data are needed and in which form the material should be organised and stored. We experience that it is too time consuming or requires expert knowledge to convert otherwise useful data into an acceptable 'resource format'. It is perhaps even more important that many tools turn out to be so complex that the goal of mastering them becomes an issue in its own right. Researchers working together with local communities on less-documented languages experience that linguistic software can be technically too demanding.

In fact, researchers in all non-computational fields of linguistics encounter problems similar to those just described for field-oriented research. Concerned with timely publication, for which linguistic data mainly takes the form of Interlinear Glosses (IG), the efficiency with which linguistic data can be created is an important issue. Several factors will affect which form linguistic data management will take, namely the standardisation of data beyond the field of NLP, non-expert user IT solutions allowing the efficient creation of linguistic data, and finally, improved availability of linguistic data for human consumption in research and publication.

4 Linguistic services and public linguistic data

Within linguistics the idea of cloud computing is relatively new: the basic concept is that users of digital technology no longer need to maintain the software they use, instead the maintenance of the technological infrastructure is left to services online. Already a success in commercial applications, IT services have also become a reality in research. Within linguistics and specifically language documentation, cloud computing facilities are at present mainly restricted to online archives. Yet, online services can be extended to provide tools for databasing and annotation of data. Scientific data exchange is an issue in biochemistry (Leser, 2009), but as far as we know it has not been an issue in linguistics. The question is not so much why we should share data but rather *how* and *what*. The linguistic tool that we would like to demonstrate gives a concrete answer to these questions. Table 1 presents a short overview of the main functionalities of the TypeCraft web application.

5 Creation, storage, migration and representation of IGs in TypeCraft

The TypeCraft web application can be used online at `http://www.typecraft.org/`. The TC wiki serves as the central hub of the application. The TC database is accessed through *My Texts* which displays the user's repository of IG collections, called *Texts*. *My Texts* is illustrated in Figure 1. Graded access is one of the design properties of TypeCraft. *My Texts* has two sections consisting of private data (data readable only be the user), and shared data. Shared data are Texts owned by groups of TC users. After being assigned to a group, the user can decide which data to share with which of his groups. Data can also be made public so that anyone on the net can read and export (but not edit) it.

TypeCraft is like the well known Linguist's Toolbox (International, 2010) an interlinear glosser. However, different from Toolbox, TypeCraft is a relational database and therefore by nature has many advantages over file-based systems like Toolbox; this not only concerns data integrity but also data migration. In addition, databases in general offer greater flexibility for search and retrieval. The other major difference between Toolbox and TypeCraft is that TypeCraft is an online service which frees the users from all the problems arising from maintaining an application on their own computer. Online databases like TypeCraft are multiuser systems, i.e. many people can access the same data at the same time independently of where they are located. Users administer their own data, either in a private domain or publicly, and they can make use of other users'

Table 1: Overview over TypeCraft Functionalities

Annotation	Collaboration	Data Migration
sentence tokenisation	graded access	manual text import
interactive table cells	tool internal user communication	export of annotated phrases to MS Word, OpenOffice.org and LaTeX
Lazy Annotation Mode	user pages for background information	XML semi-automatic export to the TC wiki
extensive search functionality	sharing of data sets between user groups	automatic update of data exported to the TC wiki

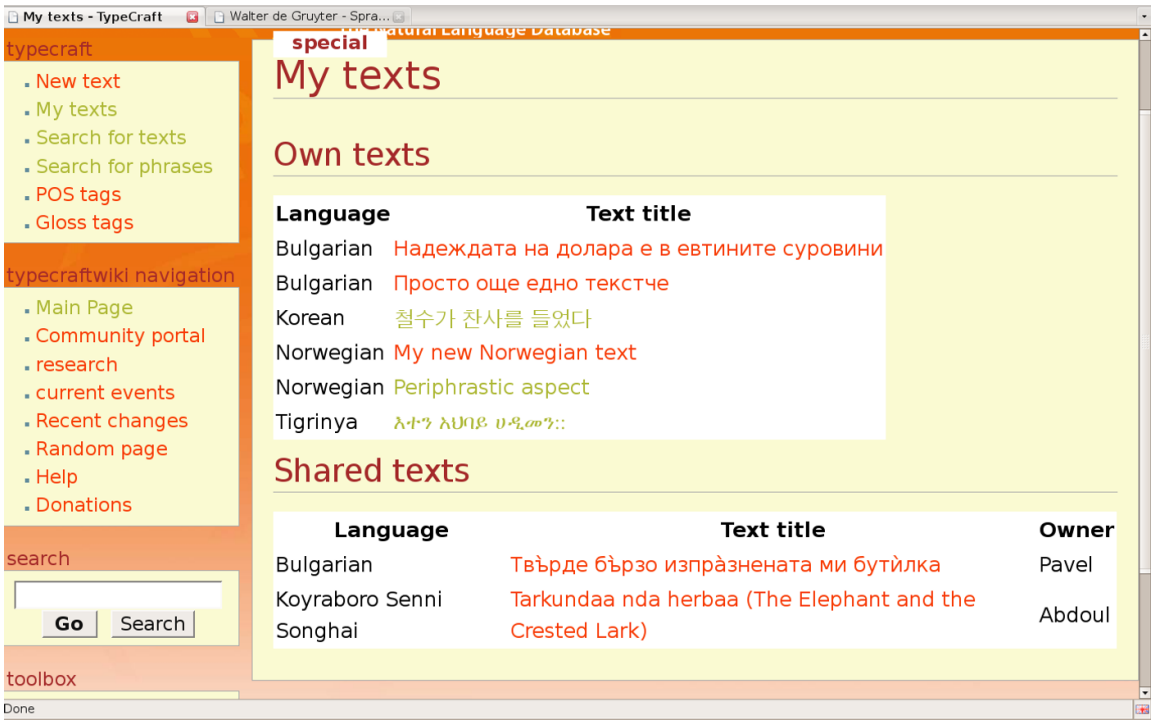


Figure 1: My texts in TypeCraft

data. Sharing information and data is an issue of mutual interest. Using standard wiki functionality, users discuss annotation issues. A TC internal email function allows users to communicate directly within the application. User pages function to personalise information and to create a TC user community. Social networking within a scientific tool plays a crucial role for the improvement of data validity. Information given by annotators, such as native language and professional background, increase the trust in TC data.

The TC wiki features interactive Google maps (a MediaWiki extension) which can be used to locate a language geographically. Isoglosses can be shown on the map too.

It is not always possible to work online. The TC online database is complemented by the TC offline client which can be downloaded from the project website for free. As a Java application it runs on multiple platforms, and allows the user to work offline in an environment familiar to him from the web application. The offline client offers the same functionality as the online service. The user can import data either locally or from the central TC database.

6 Glossing with TypeCraft

TypeCraft supports word-to-word glossing on eight tiers. After having imported a text and run it through a simple sentence splitter, the user can

click on a phrase and enter annotation mode. The system prompts the user for *lazy annotation* (in Toolbox called *sentence parsing*) which will automatically insert the annotation of already known words into the annotation table.
The user is restricted to a set of predefined tags which can be accessed from the TC wiki navigation bar where they are automatically updated when the database changes. TypeCraft is a multilingual database hosting languages from distinct language families and grammar traditions. It is therefore crucial to have standards that are extendible.

The TypeCraft tag set is mapped to the General Ontology for Linguistic Description (GOLD). GOLD (Farrar and Langendoen, 2003) has been created to facilitate a more standardised use of basic grammatical features. As an OWL ontology GOLD allows a representation of grammatical features in terms of categories and their relations. By mapping TC tags to GOLD, the user can make use of the information in the GOLD system which allows him to relate tags to more general grammatical concepts. The TypeCraft–GOLD mapping allows the user direct access to standards and necessary background information to associate glosses with the grammatical categories they are meant to express. GOLD in many cases provides definitions of concepts and important bibliographic resources related to the use of the term.

Annotated TC tokens can be exported to Microsoft Word, OpenOffice.org Writer and LaTeX. Example (1) is exported to LaTeX from TypeCraft. It illustrates locative relativisation in Runyakitara, a Bantu language spoken in Uganda:

(1)

Omu nju ei abagyenyi baataahiremu ekasya						
òmù	njù	èì	àbàgyènyì	bàátàhìrèmù		èkásyà
Omu	n ju	ei	a ba gyenyi	ba a	taah ire mu	e ka sya
in	CL9 *house*	*which*	IV CL2 *visitor*	CL2 PRS.PERF *enter*	PERF LOC	CL9 PST *burn*
PREP	N	REL	N	V		V
'The house in which visitors entered burned'						

Next to export to the main text processing systems, TypeCraft supports XML export which allows the exchange of data with other applications.

7 Conclusion

Interlinear Glosses are the most common form of linguistic data annotated by humans. In this paper we have presented an online linguistic service which allows the creation, storage and retrieval of IGs, thus granting them the status of an independent language resource. Reusability of data has become an issue also in the non-computational fields of linguistics. Although not sufficiently rewarded at the moment, already now the creation and sharing of linguistic data online is an efficient way for the creation and propagation of annotated texts in form of Interlinear Glosses. Since the TypeCraft web application provides off-the-shelf data for linguistic publications already formatted for all main text processing systems, data creation and retrieval with TypeCraft is time efficient. This makes linguistic work more data oriented and enables reasonable scientific turnover rate.

References

Bird, Steven and Gary Simons. 2003. Seven dimensions of portability for language documentation and description. *Languages*, 73(3):557–582.

Farrar, Scott and D. Terence Langendoen. 2003. A linguistic ontology for the semantic web. *GLOT International*, 7(3):97–100.

International, SIL. 2010. http://www.sil.org/, January.

Leser, Ulf. 2009. Social issues in scientific data exchange. Manuscript Humboldt Universität, Berlin.

HowNet and Its Computation of Meaning

Zhendong Dong
Research Center of Computer & Language Engineering, CAS
dzd@keenage.com

Qiang Dong
Canada Keentime Inc.
dongqiang@keenage.com

Changling Hao
Canada Keentime Inc.
support@keenage.com

Abstract

The presentation will mainly cover (1) What is HowNet? HowNet is an on-line common-sense knowledgebase unveiling inter-conceptual relationships and inter-attribute relationships of concepts as connoting in lexicons of the Chinese and their English equivalents. (2) How it functions in the computation of meaning and as a NLP platform? The presentation will show 9 HowNet-based application tools. All of them are not merely demonstration of some methodology or algorithm, but are real application tools that can be tested by users themselves. Apart from the tools that are specially designed to deal with Chinese, most of the tools are bilingual, even the WSD tool.

1 What is HowNet

HowNet is an on-line common-sense knowledgebase unveiling inter-conceptual relationships and inter-attribute relationships of concepts as connoting in lexicons of the Chinese and their English equivalents. To put it simply, relationship is the soul of HowNet, as well as the world knowledge. The relationships that represent knowledge can be divided into two categories: Concept Relationship (CR) and Attribute Relationship (AR).

It is believed that concept relationships fall into a net, which is called Concept Relation Net (CRN) and attribute relationships fall into a net too, called Attribute Relation Net (ARN). Different individual has different CRN, even of the same concept. This reflects different levels of knowledge among people. CRN is elastic or extendable as it varies with individual persons. The more knowledge one has, the more concepts he will master, and what is more, the larger or more complicated CRN of the concepts he will know. It can be imagined that a 6-year child may know "doctor" but his CRN of "doctor" would be far from that as shown in Fig. 1, which is believed to be mastered by an ordinary adult. The same case goes with mankind as a whole. Mankind increases his knowledge with each passing year when he enlarges his volume of concepts and at the same time, the CRN of the concepts.

Careful observations find that the meaning of concepts is displayed not only by its CRN but also by the relationships among attributes of the concepts, as called Attribute Relation Net. In many cases it is the attributes of a concept that act in the role of meaning representation. Fig. 2 reveals that it is not "paper" as a whole that is related to "write", but only one of its attributes, say "color", is related to "write" with "contrast" as the condition. Therefore in a strict sense, "paper" is not necessarily related to "write". We can sometimes even write on the sand with a twig or on the table with our wet finger. On the contrary, we cannot write on a piece of white paper with a chalk or on the blackboard in black ink. Therefore, for writing, what affects may not be the whole lot of the concept like "paper", but some attributes of the concept. Besides, we can use "paper" to wrap up something because of its attributes of the material, which are almost the same as cloth or plastic. HowNet is unique in its four peculiarities: (1) Use of sememes: HowNet uses sememes to interpret concepts. Sememes are regarded as the basic unit of the meaning. (2) Definition in a structuralized language: Each concept in HowNet lexicon is defined in a language, called Knowledge Database Markup Language (KDML). The KDML is mainly composed of sememes and semantic roles. The

Coling 2010: Demonstration Volume, pages 53–56,
Beijing, August 2010

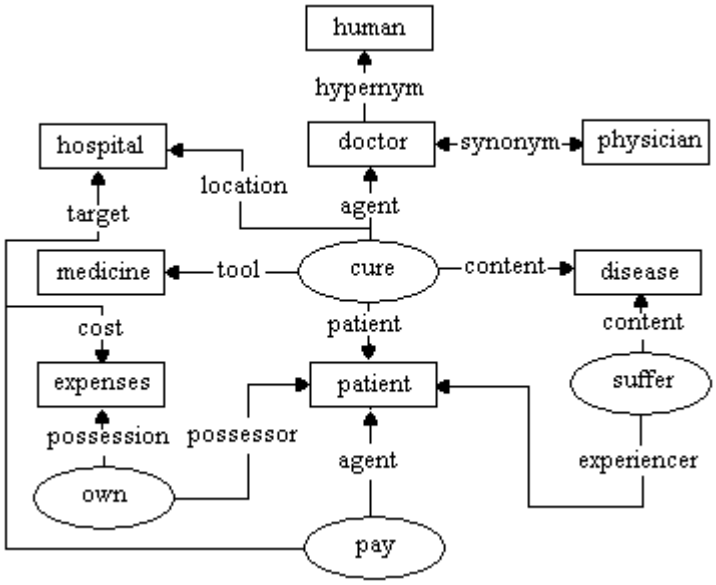


Figure 1 Concept Relation Net (CRN) of "doctor"

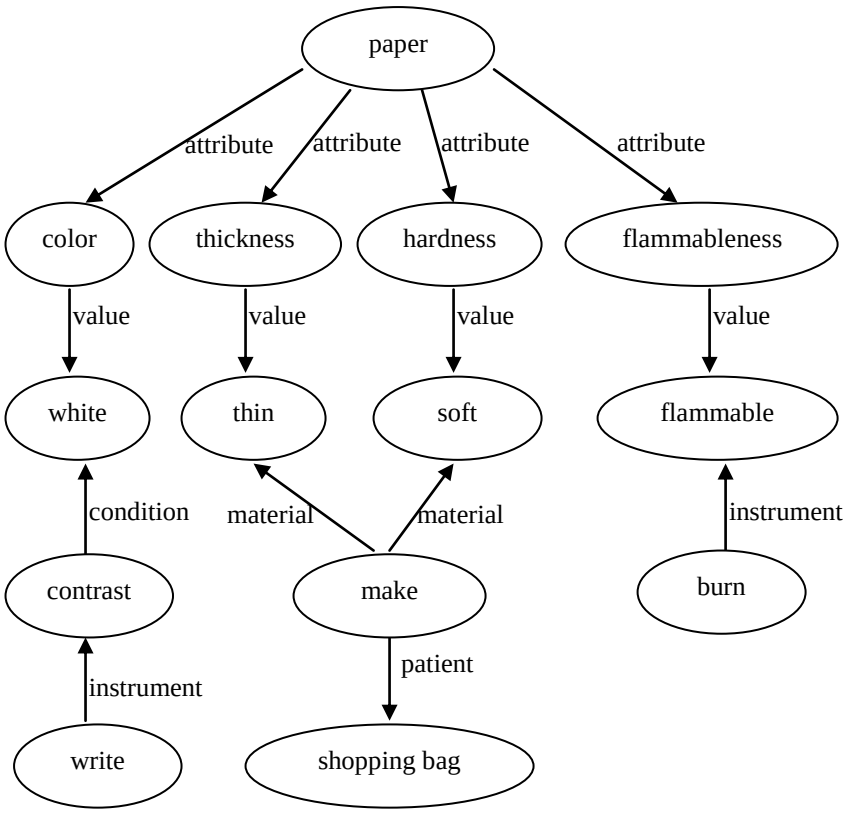


Figure 2 Attribute Relation Net (ARN) of "paper"

Knowledge Database Mark-up Language uses 2089 sememes, 128 secondary features and 94 semantic roles as its vocabulary and adopts an extended BNF as its syntax. The concept of "doctor (medical)" is defined in HowNet as:

DEF={human|人:HostOf={Occupation|职位},

domain={medical|医},{doctor|医治:agent={~}}}

All the computation of meaning in HowNet is based on the definitions of the concepts.

(3) Self-sufficiency: Systematic integration of hierarchical taxonomies, axiomatic inference, KDML-defined concepts.

(4) Language independence: In the final analysis, HowNet is not word-oriented as WordNet, but concept-oriented. Only with the HowNet's shared definitions can we achieve a shared ontology for all languages.

Table 1 shows the latest statistics of the basic data of HowNet.

Chinese Character		7182
Chinese Word & Expression		100385
English Word & Expression		96565
Chinese Meaning		115278
English Meaning		121262
Definition		30014
Record		192191
Semantics	**Chinese**	**English**
Event	14554	12881
Attribute	4351	4879
AttributeValue	10160	10140
Things	72016	72016
Time	2683	2683
Space	1244	1244
Component	8577	8577

Table 1 statistics of the basic data of HowNet

2 HowNet functions as a NLP platform

HowNet is developing toward a NLP platform. HowNet is a powerful tool for the computation of meaning. To date, 9 HowNet-based application tools have been developed. They are:

1. HowNet_Browser (E/C bilingual)
2. HowNet_Relevance (E/C bilingual)
3. HowNet_Similarity (E/C bilingual)
4. HowNet_Inference_Pool (E/C bilingual)
5. HowNet_SenseColonyTester (E/C bilingual)
6. HowNet_Translate (E-to-C)
7. HowNet_Morpho_Processor (Chinese monolingual)
8. HowNet_VN – disambiguator for Chinese V-N structure (Chinese monolingual)
9. HowNet_VXY -- disambiguator for Chinese V-N-的-N structure (Chinese monolingual)

The purpose for developing these tools is (1) to check the HowNet's data and framework for its accuracy and coverage so as to test the soundness of its philosophy and design; (2) to push HowNet near to end applications so as to provide evidence of its value as knowledge resources;

Of all these tools, HowNet Browser is the key. The Browser contains all HowNet basic data and provides various kinds of elementary or shallow computation of meanings. The basic data in HowNet can be divided into two parts: firstly, the basic lexical data and secondly taxonomies. In the lexical database, each concept is described in a fixed structure, for example,

NO.=046048
W_C=富
G_C=adj [fu4]
S_C=PlusSentiment|正面评价
E_C=~人，~婆，~国，~家子弟，~得流油，穷的穷~的~，贫~差别，先~起来，农村~了
W_E=rich
G_E=adj
S_E=PlusSentiment|正面评价
E_E=
DEF={rich|富}
RMK=

With the browser the user can retrieve all kinds of basic relations between concepts, such as synonym, hypernym, hyponym, etc. It should be noticed that these kinds of relations in HowNet are not coded manually as the way as done in WordNet, but are computed on the basis of concept definitions. The browser can give all sorts of semantic roles for a given verb concept. To take "treat" as a given event, we retrieve all its "agents", "locations", "patients", "instruments". This is regarded as the shallow relations between verb concepts and their relevant noun concepts.

Particular attention should be given to our newly developed tool, HowNet Inference Pool (E/C bilingual). With the help of an activator of the tool we can build a senses pool for any concept in HowNet. The pool covers all sorts of relationships under the key concept, for instance, when the concept of "money" as the key, it has a pool with 2600 concepts, including "bank", "deposit", "borrow", "buy", "steal", etc. Hence

suppose a question like “can we borrow money from a bank?” is raised to an inference machine, we are sure that the machine can give a correct answer with correct selection of meanings, like “bank” as “financial bank”. Moreover based on the inference machine we have developed a word sense disambiguation tool called HowNet SenseColony Tester (E/C bilingual). The tool is designed to be skilled in tackling the ambiguity of discourse type both in Chinese and English. The words “governor”, “state” in the following paragraph are so-called those of discourse-ambiguity type:

“We provided $250 in relief to more than 5 million California seniors -- many whose life savings had taken a big hit in the financial crisis. And we provided emergency assistance to our *governors* to prevent teachers and police officers and firefighters from being laid off as a result of *state* budget shortfalls. At a time when California is facing a fiscal crisis, we know that this has saved the jobs of tens of thousands of educators and other needed public servants just in this *state*. And what was true in California was true all across the country.”

The tool is language independent; it employs the data resources and the algorithm of the same type.

HowNet English-Chinese MT system is a rule-based system. It uses HowNet basic data as its English-Chinese bilingual dictionary. It is powerful in its strongly semantic basis. The system will surely have a bright future in its application to PDA products and Chinese language learning aids.

All the HowNet tools are not merely a demo of certain methodology, but are real applications that can be tested by users themselves.

References

Keh-Jiann Chen, Shu-Ling Huang, Yueh-Yin Shih, Yi-Jun Chen, 2005, Extended-HowNet: A Representational Framework for concepts, Proceedings of Second International Joint Conference 2005

Keh-Jiann Chen, 2009, E-HowNet- a Lexical Semantic Representation System and its Relation to Morphology, Syntax and Semantics, (keynote talk, at ROCLING XXI 2009)

Zhendong Dong and Qiang Dong, 2006. *HowNet and the Computation of Meaning*, World Scientific Publishing Co. Pte. Ltd., Singapore

Fellbaum, 1998, WordNet: An Electronic Lexical Datbase. Ed. Cristiane Fellbaum, The MIT Press, Cambridge, London, England, 1998.

Nagao, Makoto, 1997 Machine Translation Through Language Understanding, MT Summit VI Proceedings

Yarowsky, D. (1993) One sense per collocation. In *Proceedings, ARPA Human Language Technology Workshop*, pp. 266-271.

董振东, 董强, 2001, 知网和汉语研究, 当代语言学, 第三卷, 第1期, pp.33-44

冯志伟, 2001, 计算语言学基础, 北京, 商务印书馆.

Multiword Expressions in the wild? The mwetoolkit comes in handy

Carlos Ramisch[†*] **Aline Villavicencio**[*] **Christian Boitet**[†]
[†] GETALP – Laboratory of Informatics of Grenoble, University of Grenoble
[*] Institute of Informatics, Federal University of Rio Grande do Sul
{ceramisch,avillavicencio}@inf.ufrgs.br Christian.Boitet@imag.fr

Abstract

The mwetoolkit is a tool for automatic extraction of Multiword Expressions (MWEs) from monolingual corpora. It both generates and validates MWE candidates. The generation is based on surface forms, while for the validation, a series of criteria for removing noise are provided, such as some (language independent) association measures.[1] In this paper, we present the use of the mwetoolkit in a standard configuration, for extracting MWEs from a corpus of general-purpose English. The functionalities of the toolkit are discussed in terms of a set of selected examples, comparing it with related work on MWE extraction.

1 MWEs in a nutshell

One of the factors that makes Natural Language Processing (NLP) a challenging area is the fact that some linguistic phenomena are not entirely compositional or predictable. For instance, why do we prefer to say *full moon* instead of *total moon* or *entire moon* if all these words can be considered synonyms to transmit the idea of completeness? This is an example of a *collocation*, i.e. a sequence of words that tend to occur together and whose interpretation generally crosses the boundaries between words (Smadja, 1993). More generally, collocations are a frequent type of *multiword expression (MWE)*, a sequence of words that presents some lexical, syntactic, semantic, pragmatic or statistical idiosyncrasies (Sag et al., 2002). The definition of MWE also includes a wide range of constructions like phrasal verbs (*go ahead*, *give up*), noun compounds (*ground speed*), fixed expressions (*a priori*) and multiword terminology (*design pattern*). Due to their heterogeneity, MWEs vary in terms of syntactic flexibility (*let alone* vs *the moon is at the full*) and semantic opaqueness (*wheel chair* vs *pass away*).

While fairly studied and analysed in general Linguistics, MWEs are a weakness in current computational approaches to language. This is understandable, since the manual creation of language resources for NLP applications is expensive and demands a considerable amount of effort. However, next-generation NLP systems need to take MWEs into account, because they correspond to a large fraction of the lexicon of a native speaker (Jackendoff, 1997). Particularly in the context of domain adaptation, where we would like to minimise the effort of porting a given system to a new domain, MWEs are likely to play a capital role. Indeed, theoretical estimations show that specialised lexica may contain between 50% and 70% of multiword entries (Sag et al., 2002). Empirical evidence confirms these estimations: as an example, we found that 56.7% of the terms annotated in the Genia corpus are composed by two or more words, and this is an underestimation since it does not include general-purpose MWEs such as phrasal verbs and fixed expressions.

The goal of mwetoolkit is to aid lexicographers and terminographers in the task of creating language resources that include multiword entries. Therefore, we assume that, whenever a textual corpus of the target language/domain is available, it is possible to automatically extract interesting sequences of words that can be regarded as candidate MWEs.

2 Inside the black box

MWE identification is composed of two phases: first, we automatically generate a list of candi-

[1] The first version of the toolkit was presented in (Ramisch et al., 2010b), where we described a language- and type-independent methodology.

Coling 2010: Demonstration Volume, pages 57–60,
Beijing, August 2010

$$
\begin{aligned}
\texttt{mle} &= \frac{c(w_1 \dots w_n)}{N} \\
\texttt{dice} &= \frac{n \times c(w_1 \dots w_n)}{\sum_{i=1}^{n} c(w_i)} \\
\texttt{pmi} &= \log_2 \frac{c(w_1 \dots w_n)}{E(w_1 \dots w_n)} \\
\texttt{t-score} &= \frac{c(w_1 \dots w_n) - E(w_1 \dots w_n)}{\sqrt{c(w_1 \dots w_n)}}
\end{aligned}
$$

Figure 1: A candidate is a sequence of words w_1 to w_n, with word counts $c(w_1) \dots c(w_n)$ and n-gram count $c(w_1 \dots w_n)$ in a corpus with N words. The expected count if words co-occurred by chance is $E(w_1 \dots w_n) \approx \frac{c(w_1) \dots c(w_n)}{N^{n-1}}$.

candidate	f_{EP}	f_{google}	**class**
status quo	137	1940K	True
US navy	4	1320K	False
International Cooperation	2	1150K	False
Cooperation Agreement	188	115K	True
Panama Canal	2	753K	True
security institution	5	8190	False
lending institution	4	54800	True
human right	2	251K	True
Human Rights	3067	3400K	False
pro-human right	2	34	False

Table 1: Example of MWE candidates extracted by `mwetoolkit`.

dates from the corpus; then we filter them, so that we can discard as much noise as possible. *Candidate generation* uses flat linguistic information such as surface forms, lemmas and parts of speech (POS).[2] We can then define target sequences of POS, such as `VERB NOUN` sequences, or even more fine-grained constraints which use lemmas, like *take* `NOUN` and *give* `NOUN`, or POS patterns that include wildcards that stand for any word or POS.[3] The optimal POS patterns for a given domain, language and MWE type can be defined based on the analysis of the data.

For the *candidate filtering* a set of association measures (AMs), listed in figure 1, are calculated for each candidate. A simple threshold can subsequently be applied to filter out all the candidates for which the AMs fall below a user-defined value. If a gold standard is available, the toolkit can build a classifier, automatically annotating each candidate to indicate whether it is contained in the gold standard (i.e. it is regarded as a true MWE) or not (i.e. it is regarded as a non-MWE).[4] This annotation is not used to filter the lists, but only by the classifier to learn the relation between the AMs and the MWE class of the candidate. This is particularly useful because, to date, it remains unclear which AM performs better for a particular type or language, and the classifier applies measures according to their efficacy in filtering the candidates.Some examples of output are presented in table 1.

3 Getting started

The toolkit is open source software that can be freely downloaded (`sf.net/projects/mwetoolkit`). As a demonstration, we present the extraction of noun-noun compounds from the general-purpose English Europarl (EP) corpus[5].

To preprocess the corpus, we used the sentence splitter and tokeniser provided with EP, followed by a lowercasing treatment (integrated in the toolkit), and lemmatisation and POS tagging using the TreeTagger[6]. The tagset was simplified since some distinctions among plural/singular and proper nouns were irrelevant.

From the preprocessed corpus, we obtained all sequences of 2 nouns, which resulted in 176,552 unique noun compound candidates. Then, we obtained the corpus counts for the bigrams and their component unigrams in the EP corpus. Adopting the web as a corpus, we also use the number of pages retrieved by Google and by Yahoo! as

[2] If tools like a POS tagger are not available for a language/domain, it is possible to generate simple n-gram lists ($n = 1..10$), but the quality will be inferior. A possible solution is to filter out candidates on a keyword basis, e.g. from a list of stopwords).

[3] Although syntactic information can provide better results for some types of MWEs, like collocations (Seretan, 2008), currently no syntactic information is allowed as a criterion for candidate generation, keeping the toolkit as simple and language independent as possible.

[4] The gold standard can be a dictionary or a manually annotated list of candidates.

[5] `www.statmt.org/europarl`.

[6] `http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/`.

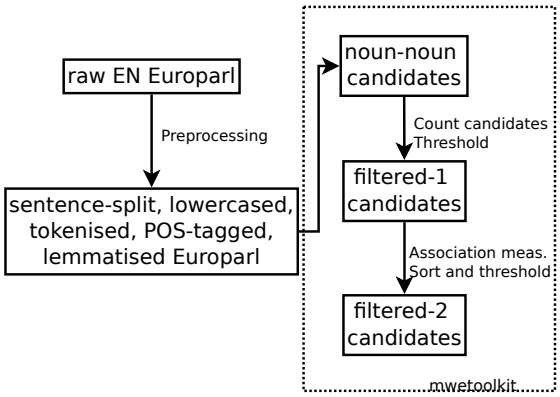


Figure 2: Step-by-step demonstration on the EP corpus.

counts. The `mwetoolkit` implements a cache mechanism to avoid redundant queries, but to speed up the process[7], we filtered out all candidates occurring less than two times in EP, which reduced the list of candidates to 64,551 entries (*filtered-1 candidates* in figure 2).

For the second filtering step, we calculated four AMs for each of the three frequency sources (EP, Google and Yahoo!). Some results on machine learning applied to the candidate lists of the `mwetoolkit` can be found in Ramisch et al. (2010b). Here, we will limit ourselves to a discussion on some advantages and inconvenients of the chosen approach by analysing a list of selected examples.

4 Pros and cons

One of the biggest advantages of our approach is that, since it is language independent, it is straightforward to apply it on corpora in virtually any language. Moreover, it is not dependent on a specific type of construction or syntactic formalism. Of course, since it only uses limited linguistic information, the accuracy of the resulting lists can always be further improved with language-dependent tools. In sum, the toolkit allows users to perform systematic MWE extraction with consistent intermediary files and well defined scripts and arguments (avoiding the need for a series of ad hoc separate scripts). Even if some basic knowledge about how to run Python scripts and how to pass arguments to the command line is necessary, the user is not required to be a programmer.

Nested MWEs are a problem in the current approach. Table 1 shows two bigrams *International Cooperation* and *Cooperation Agreement*, both evaluated as False candidates. However, they could be considered as parts of a larger MWE *International Cooperation Agreement*, but with the current methodology it is not possible to detect this kind of situation. Another case where the candidate contains a MWE is the example *pro-human right*, and in this case it would be necessary to separate the prefix from the MWE, i.e. to re-tokenise the words around the MWE candidate. Indeed, tools for consistent tokenisation, specially concerning dashes and slashes, could improve the quality of the results, in particular for specialised corpora.

The toolkit provides full integration with web search engine APIs. The latter, however, are of limited utility because search engines are not only slow but also return more or less arbitrary numbers, some times even inconsistent (Ramisch et al., 2010c). When large corpora like EP are available, we suggest that it is better to use its counts rather than web counts. The toolkit provides an efficient indexing mechanism, allowing for arbitrary *n*-grams to be counted in linear time.

The automatic evaluation of the candidates will always be limited by the coverage of the reference list. In the examples, *Panama Canal* is considered as a true MWE whereas *US navy* is not, but both are proper names and the latter should also be included as a true candidate. The same happens for the candidates *Human Rights* and *human right*. The `mwetoolkit` is an early prototype whose simple design allows fine tuning of knowledge-poor methods for MWE extraction. However, we believe that there is room for improvement at several points of the extraction methodology.

5 From now on

One of our goals for future versions is to be able to extract bilingual MWEs from parallel or comparable corpora automatically. This could be done through the inclusion of automatic word alignment information. Some previous experiments show, however, that this may not be enough, as

[7] Yahoo! limits the queries to 5,000/day.

automatic word alignment uses almost no linguistic information and its output is often quite noisy (Ramisch et al., 2010a). Combining alignment and shallow linguistic information seems a promising solution for the automatic extraction of bilingual MWEs. The potential uses of these lexica are multiple, but the most obvious application is machine translation. On the one hand, MWEs could be used to guide the word alignment process. For instance, this could solve the problem of aligning a language where compounds are separate words, like French, with a language that joins compound words together, like German. In statistical machine translation systems, MWEs could help to filter phrase tables or to boost the scores of phrases which words are likely to be multiwords.Some types of MWE (e.g. collocations) could help in the semantic disambiguation of words in the source language. The sense of a word defined by its collocate can allow to chose the correct target word or expression (Seretan, 2008).

We would also like to improve the techniques implemented for candidate filtering. Related work showed that association measures based on contingency tables are more robust to data sparseness (Evert and Krenn, 2005). However, they are pairwise comparisons and their application on arbitrarily long *n*-grams is not straightforward. An heuristics to adapt these measures is to apply them recursively over increasing *n*-gram length. Other features that could provide better classification are context words, linguistic information coming from simple word lexica, syntax, semantic classes and domain-specific keywords. While for poor-resourced languages we can only count on shallow linguistic information, it is unreasonable to ignore available information for other languages. In general, machine learning performs better when more information is available (Pecina, 2008).

We would like to evaluate our toolkit on several data sets, varying the languages, domains and target MWE types. This would allow us to assign its quantitative performance and to compare it to other tools performing similar tasks. Additionally, we could evaluate how well the classifiers perform across languages and domains. In short, we believe that the `mwetoolkit` is an important first step toward robust and reliable MWE treatment. It is a freely available core application providing flexible tools and coherent up-to-date documentation, and these are essential characteristics for the extension and support of any computer system.

Acknowledgements

This research was partly supported by CNPq (Projects 479824/2009-6 and 309569/2009-5), FINEP and SEBRAE (COMUNICA project FINEP/SEBRAE 1194/07).

References

Evert, Stefan and Brigitte Krenn. 2005. Using small random samples for the manual evaluation of statistical association measures. *Comp. Speech & Lang. Special issue on MWEs*, 19(4):450–466.

Jackendoff, Ray. 1997. Twistin' the night away. *Language*, 73:534–559.

Pecina, Pavel. 2008. Reference data for czech collocation extraction. In *Proc. of the LREC Workshop Towards a Shared Task for MWEs (MWE 2008)*, pages 11–14, Marrakech, Morocco, Jun.

Ramisch, Carlos, Helena de Medeiros Caseli, Aline Villavicencio, André Machado, and Maria José Finatto. 2010a. A hybrid approach for multiword expression identification. In *Proc. of the 9th PROPOR (PROPOR 2010)*, volume 6001 of *LNCS (LNAI)*, pages 65–74, Porto Alegre, RS, Brazil. Springer.

Ramisch, Carlos, Aline Villavicencio, and Christian Boitet. 2010b. mwetoolkit: a framework for multiword expression identification. In *Proc. of the Seventh LREC (LREC 2010)*, Malta, May. ELRA.

Ramisch, Carlos, Aline Villavicencio, and Christian Boitet. 2010c. Web-based and combined language models: a case study on noun compound identification. In *Proc. of the 23th COLING (COLING 2010)*, Beijing, China, Aug.

Sag, Ivan, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *Proc. of the 3rd CICLing (CICLing-2002)*, volume 2276/2010 of *LNCS*, pages 1–15, Mexico City, Mexico, Feb. Springer.

Seretan, Violeta. 2008. *Collocation extraction based on syntactic parsing*. Ph.D. thesis, University of Geneva, Geneva, Switzerland.

Smadja, Frank A. 1993. Retrieving collocations from text: Xtract. *Comp. Ling.*, 19(1):143–177.

Author Index

Akegawa, Naoki, 1
Anders, Björkelund, 33

Baldwin, Timothy, 37
Beermann, Dorothee, 49
Bernd, Bohnet, 33
Besana, Sveva, 17
Boitet, Christian, 57

Che, Wanxiang, 13
Chen, Keh-Jiann, 45
Chen, Wei-Te, 45
Chung, You-Shan, 45
Colowick, Susan, 37

Dong, Qiang, 53
Dong, Zhendong, 53

Fabbrizio, Giuseppe, 17

Gupta, Narendra, 17

Hafdell, Love, 33
Han, Dongli, 1
Hao, Changling, 53
Hori, Tomomasa, 1
Hsieh, Shu-Kai, 5
Huang, Shu-Ling, 45

Jin, Wei, 29

Kamochi, Shuntaro, 1

Li, Zhenghua, 13
Lin, Su-Chu, 45
Liu, Ting, 13
Loh, Stanley, 21

Mani, Premkumar, 17
Mihaylov, Pavel, 49
Miltsakaki, Eleni, 41
Mírovský, Jiří, 9
Mladová, Lucie, 9
Muller, Daniel, 21

Nugues, Pierre, 33

Pool, Jonathan, 37

Ramisch, Carlos, 57

Silva, Fabio, 21
Song, Xin, 1

Tian, Yan, 25

Villavicencio, Aline, 21, 57

Wilkens, Rodrigo, 21
Wives, Leandro, 21
Wu, Xin, 29
Wu, Yueh-Cheng, 5

Žabokrtský, Zdeněk, 9