Cornetto Deliverable D08: Acquisition Toolkit

Erik Tjong Kim Sang
University of Amsterdam
erikt@science.uva.nl

April 28, 2008

1 Introduction

This document describes deliverable D08 of the Stevin project Cornetto. The deliverable contains the so called *acquisition toolkit*, the software which have been developed at the University of Amsterdam for extracting lexical relations from text.

2 Installation and usage summary

In order to install the software you need to unpack the archive file and run make in the software directory:

```
$ unzip D08.zip
$ cd D08
$ make
```

The software contains different methods for extracting hypernym information. Most methods need a plain text input file. In the examples we use the file wiki.txt with approximately one million words of the Dutch Wikipedia.

First, we use zoals (such as) patterns for finding hypernyms:

```
$ bin/run.zoals wiki.txt
1. Preprocessing text file $FILE (this might take a while)..."
2. Extracting zoals patterns...
3. Combining zoals patterns...
4. Evaluating...
precision: 0.18182; recall: 0.00022; f: 0.00043; dist: 2.30
5. Done. Hypernym pairs have been stored in wiki.insert.zoals
```

Next, we use conjunction (and/or) patterns for extracting hypernyms:

```
$ bin/run.conj wiki.txt
1. Preprocessing text file $FILE (this might take a while)..."
2. Extracting conjunction patterns...
3. Combining conjunction patterns...
4. Evaluating...
precision: 0.03686; recall: 0.00148; f: 0.00284; dist: 1.43
5. Done. Hypernym pairs have been stored in wiki.insert.conj
```

We can also use examples of hypernym pairs to find extraction patterns and then apply these patterns for finding hypernyms:

```
$ bin/run.bbr wiki.txt
1. Preprocessing text file $FILE (this might take a while)..."
2. Extracting noun pairs...
3. Building training file for machine learner...
4. Running machine learner...
5. Evaluating...
precision: 0.31105; recall: 0.00233; f: 0.00462; dist: 1.70
6. Done. Hypernym pairs have been stored in wiki.insert.bbr
```

Finally, we apply a morphological rule to the data:

```
$ bin/run.morph wiki.txt
1. Preprocessing text file $FILE (this might take a while)..."
2. Applying morphological rule to nouns...
3. Evaluating...
precision: 0.44865; recall: 0.04665; f: 0.08452; dist: 1.18
4. Done. Hypernym pairs have been stored in wiki.insert.morph
```

The four methods have been described in the paper *Automatic Extraction of Dutch Hypernym-Hyponym*, by Erik Tjong Kim Sang and Katja Hofmann (Proceedings of CLIN-2006, LOT Occasional Series, Utrecht, pages 163-174, 2007.

3 Background information

This software collection contain programs for extracting hypernym-hyponym pairs. A hypernym of a word A is another word B which both covers the meaning of word A and has a broader meaning. For example *furniture* is a hypernym of *table*. If word B is a hypernym of word A then A is a hyponym of B. So *table* is a hyponym of *furniture*.

3.1 Using fixed patterns for finding hypernyms

This software can extract hypernym-hyponym pairs from text. The standard method for achieving this is to look for fixed phrases, for example two nouns appearing together with *such as*: *furniture such as tables*. This has been implemented as the program `bin/run.zoals` which search for phrases containing the Dutch equivalent of *such as*: *zoals*. The task is performed in four steps:

1. preprocessing: identify words and sentences in the input text, and assign classes and lemmas to the words
2. collecting all phrases *Noun zoals Noun* and *Noun zoals Noun, ... en Noun*
3. combining all phrases for each hyponym and selecting the most likely hypernym
4. evaluation: compare the selected hyponym-hypernym pairs with the hypernyms in the Dutch section of EuroWordNet

The input FILE should be a plain text file containing Dutch text. An example test file with text from the Dutch Wikipedia (wiki.txt) has been included in this distribution.The achieved performance will be displayed at the screen and the extracted pairs will be stored in the file FILE.insert.zoals, where FILE is the name of the input file without extension. In this file, each word pair appear on a separate line with the format: number hypernym hyponym. The number has no particular meaning.

3.2 Using conjunctions for finding hypernyms

A second method for finding hypernyms is by looking for phrases like *tables and chairs*. These denote a sibling relation between *table* and *chair*. If *table* is an unknown word while *chair* is a known word with *furniture* as hypernym then *furniture* is probably also a good candidate hypernym for *table*. This idea is implemented in the program `bin/run.conj` which also performs four steps:

1. preprocessing: identify words and sentences in the input text, and assign classes and lemmas to the words
2. collecting all phrases *Noun en/of Noun* and *Noun, ..., Noun en/of Noun*
3. combining all phrases for each hyponym and selecting the most likely hypernym based on the information in the Dutch section of EuroWordNet
4. evaluation: compare the selected hyponym-hypernym pairs with the hypernyms in the Dutch section of EuroWordNet

As with the *zoals* package, the achieved performance will be displayed at the screen. The extracted pairs will be stored in the file FILE.insert.conj, where FILE is the name of the input file without extension. The required input FILE format is the same as for run.zoals. Since both packages perform the same preprocessing step, it is possible to skip this step by invoking the software with a preprocessed file with extension `.lem`: `bin/run.conj FILE.lem`.

3.3 Discovering new patterns from examples

It is difficult to foresee all patterns in which hypernym-hyponym pairs can appear. We have also implemented software which searches for such patterns based on the known pairs in the lexical resource, the Dutch part of EuroWordNet. For example, if we know that *furniture* is a hypernym of *table* and we discover the phrase *tables and other furniture* then *Noun and other Noun* is a good candidate phrase for finding other hypernyms.

`bin/run.bbr` collects all phrases up to four words between two known hypernyms. For each such phrase, it keeps four variants: which include or exclude the word just before the first noun and the word just behind the final noun. All the combinations of noun pairs and phrases are fed to a machine learner (we use Bayesian Logistic Regression) which identifies the best patterns and uses them for finding new hypernym-hyponym pairs.

`bin/run.bbr` performs five steps:

1. preprocessing: identify words and sentences in the input text, and assign classes and lemmas to the words
2. collecting all the noun pairs and interesting phrases from the text
3. combining all information for each noun pair and creating a training file for the machine learner

4. running the machine learner with 10-fold cross validation: use 90of file for training and 10with different test sections)

5. evaluation: compare the selected hyponym-hypernym pairs with the hypernyms in the Dutch section of EuroWordNet

The input format requirements and the output format are the same as for the two previously discussed programs, `bin/run.zoals` and `bin/run.conj`. Its output is stored in the file FILE.insert.bbr.

3.4 Using word-internal information

Dutch contains many compounds like *politieauto* (police car) and often the final part of the compound (here: *auto* or *car*) is a good candidate hypernym for the compound word. The final program *bin/run.morph* makes use of this feature. It tries to split input nouns and when if finds a valid minimal suffix of at least three characters which is also a noun, it outputs this one as a candidate hypernym.

The program performs three steps:

1. preprocessing: identify words and sentences in the input text, and assign classes and lemmas to the words

2. applying the reduction rule (also called morphological rule) to all the nouns in the text

3. evaluation: compare the selected hyponym-hypernym pairs with the hypernyms in the Dutch section of EuroWordNet

The input format requirements and the output format are the same as for the three previously discussed programs. Program output is stored in the file FILE.insert.morph.

3.5 Evaluation

The output of the four programs is compared with the Dutch section of the lexical resource EuroWordNet. Only words that are present in the resource are used in the evaluation. No attempt is made to check new or unknown words. Evaluation involves measuring precision, recall and F rates over identified word pairs. Recall scores are extremely low because the target is to find hypernyms of all the words in the lexical resource, an impossible task with any amount of text.

A fourth measure, distance, has been included in the evaluation process. It measures the average distance between words and correctly identified hypernyms in EuroWordNet hypernymy hierarchy. The ideal distance is one. The distance measure has been included in order to be able to check for methods which simply assign EuroWordNet's top hypernym (*something*) to every noun.

The four methods and their performances are extensively discussed in the paper *Automatic Extraction of Dutch Hypernym-Hyponym*, by Erik Tjong Kim Sang and Katja Hofmann (Proceedings of CLIN-2006, LOT Occasional Series, Utrecht, pages 163-174, 2007). The performances achieved on the supplied test section of Wikipedia give a good idea about the strengths of the four methods:

```
zoals: P: 0.18182; R: 0.00022; F: 0.00043; dist: 2.30; total: 55
 conj: P: 0.03686; R: 0.00148; F: 0.00284; dist: 1.43; total: 1845
  bbr: P: 0.28947; R: 0.00215; F: 0.00427; dist: 1.71; total: 342
morph: P: 0.44865; R: 0.04665; F: 0.08452; dist: 1.18; total: 4781
```

(here total represents the total number of correct pairs) Fixed patterns (zoals) have a reasonable recall but their recall is low. Conjunctions (conj) find more pairs but their precision is low. Pattern combination (bbr) outperforms both of them. Word-internal information (morph) achieves the highest scores but it only handle compounds. Neither of the methods performs well enough to use as a stand-alone lexicon expansion method. Additional manual checks of the output are required.

4 Applying the software for extracting other relations

The part of the software that uses the machine learner can be adapted to derive other relations than hypernymy. In order to achieve this, you only need to replace the file with hypernymy examples with examples of another relation.

In the example file, each word pair appears on a separate line. Before the pair there is a number which denotes the distance in a tree between the two words. Use 11 for distance 1, 12 for distance 2 and so on. If the relation is flat then 11 should be used on all lines. The position of the word on the line is important. If the relation has a direction then all words of the same level should be in the same column (either second or third on the line). We have not tested deriving symmetric relations.

We have tested deriving one extra relation: the capital-country relation. We have created a small example file (30 lines) which includes both capital-country pairs (11 Tokio

Japan) as well as capital-country adjective pairs (11 Tokio Japanese). After replacing `etc/examples` with this file, the machine learner extracted three capital-country pairs from the Wikipedia test file, all of which were correct (recall was 0.07895).

5 Other programs

This software package contains two more programs which could be useful:

bin/preprocess text preprocessor which can be called separately. It identifies words and sentences in text files, and assigns classes and lemmas to the words in the text

Example call: bin/preprocess ¡ FILE

Output printed on screen, one word per line (word class lemma):

Hij VNW(pers,pron,nomin,vol,3,ev,masc) Hij
heeft WW(pv,tgw,met-t) hebben
een LID(onbep,stan,agr) een
zoon N(soort,ev,basis,zijd,stan) zoon

yahoo/bin/yahooNP searches on Yahoo! for a particular phrase and returns all nouns following (or preceding) the phrase. Can be used for looking for hypernym candidates on the web.

Example call: yahoo/bin/yahooNP "zoals tafels" reverse

Output printed on screen (frequency word):
...
4 meubilair
6 meubelen
6 voorwerpen
8 meubels

When invoked without the keyword `reverse` the program will look for words following the phrase (*tables and other X*).

6 Platform

The software in this package will only work on Linux systems. Many of the parts should work at other Unix platforms like MacOSX. However, the machine learners included here (Bayesian Logistic Regression for pattern combination and the TnT tagger for preprocessing) are precompiled binaries which cannot be run on other platforms.

7 References

Thorsten Brants, *TnT – A Statistical Part-of-Speech Tagger*. Saarland University, 1999. `http://www.coli.uni-saarland.de/~thorsten/tnt/` Alexander Genkin, David D. Lewis, and David Madigan, *BBR: Bayesian Logistic Regression Software*. Rutgers University, NJ, 2005. `http://www.stat.rutgers.edu/~madigan/BBR/` Erik Tjong Kim Sang and Katja Hofmann, Automatic Extraction of Dutch Hypernym-Hyponym. In *Proceedings of CLIN-2006*, LOT Occasional Series, Utrecht, pages 163-174, 2007.

Piek Vossen, *EuroWordNet: A Multilingual Database with Lexical Semantic Networks*. Kluwer Academic Publisher, 1998.