The Språkdata-ML System as Used for SENSEVAL-2

Dimitrios KOKKINAKIS
Språkdata, Göteborg University
Box 200, SE-405 30
Göteborg, Sweden
Dimitrios.Kokkinakis @svenska.gu.se

Abstract

This paper describes the Språkdata-ML system as used in the SENSEVAL-2 exercise. The main focus of the paper is devoted to the process of feature extraction, preparation and organization of the test and training data.

Introduction

The methodology followed for sense disambiguation of the Swedish data by the Språkdata-ML system is supervised, based on Machine Learning (ML) techniques, particularly Memory Based Learning (MBL). The MBL implementation we used originates from the university of Tilburg in a system called TiMBL; details can be found in Daelemans *et al.* (1999). Thus, our main contribution in this task has been the effort to try and isolate a set of features that could maximize the performance of the MBL software. However, it is rather difficult to give the exact number of features and examples required for an adequate description of a word's sense or which algorithm performs best. We think that there is space for improvement of our system's performance by better modeling of the available resources (e.g. context, annotations), choice of parameters and algorithms, a claim that we have not explored to its full potential, further exploration is required. Intelligent example selection for supervised learning is an important issue in ML, an issue that we have not fully explored. In previous experiments for a similar problem for Swedish, the algorithm that performed best in TiMBL was a variant of the *k-nearest neighbor* (Mitchell, 1997) called IB1, an algorithm that we also used in the exercise; (Kokkinakis & Johansson Kokkinakis, 1999).

1 Data Preparation (Train)

To enhance the lexical disambiguation results using the available resources, we perform pre-processing in both the dictionary and the text to be sense-disambiguated. This is motivated by the fact that by making certain normalizations and simplifications in the resources we (hopefully) contribute to the production of qualitatively better results.

Initially, a text to be disambiguated is pre-processed by a tokeniser, a sentence boundary identifier, an idiom[1] and multiword identifier, a Name-Entity recogniser[2], a part-of-speech tagger, a lemmatiser and a semantic tagger[3]. Then, the input texts are transformed to the specified format that the MBL requires, which is feature-vectors of a specific length and content. The vectors we use consist of 102 features, the last two being the *id-number* and *class* or *sense* assigned to the vector. Since we do not know in advance which features will be useful for each particular word and sense, we chose to include features from a number of different information sources.

2 Vector Creation

The vectors consisted of: (i) selected information gathered from the dictionary entries (5 features); (ii) near-context (5 features); (iii) annotations applied on the training corpus (5

[1] The idioms originate from the Gothenburg Lexical Data Base/Semantic Database (GLDB/SDB) (http://spraakdata.gu.se/lb/gldb.html) and were used for the recognition and marking of idioms in the test/training corpus (over 4,000 idioms).

[2] See http://spraakdata.gu.se/svedk/ne.html for a demo.

[3] The semantic tagger originates from work by Kokkinakis *et al.* (2000) and uses the SIMPLE semantic classes for annotation (only nouns).

features); and (iv) information acquired from the lemmatised training corpus (85 features).

The corpus instances and dictionary were in XML format. An example of a corpus instance (1) for the first sense of the noun barn 'child' and a fragment of its dictionary description (2) are:

(1) **<instance id="barn.114"><answer instance="barn.114" senseid="barn_1_1" /> <context>**... försöken så att spädbarnen själva kunde styra de retningar som de utsattes för under försöket. Inom språkforskningen betyder det att **<head>**barnen**</head>** kan påverka hur olika talljud presenteras. När de får ... **</context> </instance>**

(2) <***lemma-entry*** id="barn_1" form= "barn" pos="n" inflection="~et =""><***lexeme*** id= "barn_1_1"><definition> människa som ej vuxit färdigt</definition> <definition-ext>till kropp och själ; under ngn åldersgräns som beror på samman-hanget</definition-ext> <synt-example>kvinnor och ~ släpptes fria </synt-example><synt-example>~ under 6 år kommer in gratis</synt-example><compound>spädbarn</compound>...<***cycle*** id=" barn_1_1_a"><trans>spec. om människa som ej nått pubertetsålder, straff-myndighetsålder etc.</trans><synt-example> ännu något år är hon ett ~</synt-example><compound> barnarbete </compound><compound>barnavårdsnämnd</compound></***cycle***>...</***lexeme***><***lexeme***>...<***cycle*** id=" barn_1_2_a"> <trans> äv. utvidgat, spec. om foster </trans><synt-example>hon är med ~ </synt-example><valency>med ~ </valency> </***cycle***>...</***lexeme***></***lemma-entry***>

2.1 Vector Creation (Dictionary)

The modeling of the vectors was performed in stages. The first stage of the processing uses the information from the dictionary. For every sense and sub-sense we extracted five representative nouns from the definition (and the definition extension) by applying part-of-speech tagging, lemmatization and exclusion of a number of *generic* nouns from a stop-list e.g. människa 'human' (a). If the number of nouns were less than five, we completed the list with compounds (if available).

Furthermore, the syntactic examples were used as training corpus and were added to the training instances (b). The valency information (if any) was also used in the same way (c). Consequently the amount of training material increased with 1,296 "new" disambiguated instances. A "dummy" *XXX* instance-number was given in these cases.

We did not put much effort on a more complex processing of the definitions since these are very short. The representations given below use the dictionary and corpus sample provided in (1) and (2).

(a) **<definition>**människa som ej vuxit färdigt**</definition><definition-ext>**till kropp och själ; under ngn åldersgräns som beror på sammanhanget **</definition-ext>**
become: barn_1_1: kropp, själ, åldersgräns

(b) **<synt-example>**kvinnor och ~ släpptes fria**</synt-example>**
become: **<instance id="barn.XXX"> <answer instance="barn.XXX" senseid=** "barn_1_1"**/> <context>** kvinnor och **<head>**barn**</head>** släpptes fria **</context></instance>**

(c) **<valency>**med ~**</valency>**
become: **<instance id="barn.XXX"> <answer instance="barn.XXX" senseid=** "barn_1_2_a"**/><context>** med **<head>**barn **</head> </context></instance>**

2.2 Vector Creation (Near Context)

The second stage involved the use of the near-context. Punctuation, auxiliary verbs and a number of other stop-words were removed and the surrounding tokens (±2) of each headword in the corpus were extracted (d). Only the lemma form of the headwords was used, and the context was not lemmatized:

(d) **<instance id="barn.114"><answer instance="barn.114" senseid=** "barn_1_1" **/><context>**... språkforskningen betyder ~~det att~~ **<head>**barn~~en~~ **</head>** ~~kan~~ påverka ~~hur~~ olika ...**</context> </instance>**
became: **<instance id="barn.114"> <answer instance="barn.114" senseid="** barn_1_1"**/><context>**språkforskningen betyder **<head>**barn**</head>** påverka olika **<context></instance>**

2.3 Vector Creation (Global Features)

During the third stage, the training corpus was processed by a name-entity recognizer (e.g. HUMAN, TIME), an idiom identifier (IDIOM) and a semantic tagger (e.g. BIO, ETHNOS, PHENOMENON). The annotations produced by these tools were gathered in the form of a list of labels, and the five most frequent in the respective set of instances for each sense and sub-sense were used in the vectors. For example, for the sense barn_1_1 the five most frequent annotations found in all training instances were: BIO, ORGANIZATION-AGENCY, LOCATION, SITU and OCCUPATION-AGENT.

2.4 Vector Creation (Global Context)

Often, near-context cannot distinguish between different senses. In such cases it is useful to look at a larger context and extract keywords representative for each sense. We made a frequency list of all noun and verb occurrences for all corpus instances for each sense. From the produced lists, 85 keywords per sense were extracted by eliminating high frequency (a word occurred in more than *X* percent of the cases with the sense) and low frequency words (a word occurred at least *Z* times in the list). For the sense barn_1_1 the 85 keywords included:

> ansikte, ansvar, apparatur, arm, avvikelse, barnmorska, barnomsorg, beredskap, betala, bild, detalj, dialog, djur, docka, erfarenhet, fel, föreställning, förslag, ...

After the collection and combination of the 95 features common to a sense (stages i, iii, iv in Section 2, e1), a complete case for a sense was produced (e2):

(e1) *Lemma_SENSE: 5 words from the dictionary information, 5 "semantic" labels, 85 representative words from the global context*

(e2) barn_1_1: kropp, själ, småbarn, spädbarn, åldersgräns, BIO, ORGANIZATION-AGENCY, LOCATION, SITU, OCCUPATION-AGENT, ansikte, ansvar, apparatur, arm, avvikelse, barnmorska, barnomsorg, ...

We assume then, that for each training instance the above list is "true" and we convert the training instances into vectors of 102 features, where the 95 positions of the features in each vector were substituted with '1' keeping intact the near context. Thus, the truncated training instance in (f) was re-formatted to (g):

(f) **<instance id="barn.114"><answer instance="barn.114" senseid= "barn_1_1" /><context>**språkforskningen betyder **<head>**barn**</head>** påverka olika **<context></instance>**

(g) språkforskningen, betyder,**<head>**barn **</head>**, påverka, olika, 1, 1, 1, 1, 1, 1, 1,..., barn.114, barn_1_1.

3 Data Preparation (Test)

The test material consisted of 1,525 corpus instances in the same format as the previous training example, but without any designation of the correct *senseid*. The material was processed in a similar manner as the training one. The major difference lies in the fact that at the vector-creation stage we used the feature-vectors representative for a sense, example (e) previously, and we compared them with the features produced for each test instance. A feature at a specific position then was assigned '1' if the feature in the test occurred in the representative feature vector or '0' otherwise. For instance, the test instance in (h) was transformed, after processing, to a 102-feature-vector.

(h) **<instance id="barn.114"><answer instance="barn.114" senseid= "??????" /><context>**I jungfrukammaren innanför köket bodde en kokerska och en husa. [Ett hus fyllt av minnen] Huset är fyllt av minnen. I fotoalbumen kan vi se farmor omgiven av sina små vitklädda **<head>**barn**</head>** och pappa i sjömanskostym lutad mot en björk. I farfars svarta, snidade skrivbord ... **</context> </instance>**

The class of the representative sense-vector that produced more '1's for the test instance was chosen as the class of that instance. In (i) there are four '1's which means that the specific test instance had four common features with the representative vector for sense barn_1_2_a, and less than four for all the other representative vectors for the rest of the senses for barn. Thus, the class for the test instance is assigned that sense (which may be altered by the MBL software during the nearest-neighbor

calculation). Thus, the test instance in (h) was transformed to the format illustrated in (i). The four '1's denote that there were four features in common with the representative vector for barn_1_2_a, the rest of the representative sense-vectors for barn (e.g. barn_1_1_a, barn_1_1_b etc.) had less common features than four, and so barn_1_2_a was chosen:

(i) små, vitklädda, <head>barn</head>, pappa, i, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, barn.114, barn_1_2_a

The training and test feature vectors were then fed to the TiMBL software, where the IB1 algorithm (nearest neighbor search) was used.

4 Results

Table 1 shows the evaluation of the test material. Since answers were provided for the whole material, precision and recall obtain the same value. Coarse-grain evaluation was not used, however coarse-grained is considered the least interesting of the three measures.

	INSTANCES	FINE	MIXED
ADJECTIVES	191	48,2%	54,4%
NOUNS	616	71,3%	74,9%
VERBS	718	57,8%	66,1%
MOST FREQ. BASELINE	45,3%		
WHOLE SAMPLE	**1,525**	**62,0%**	**68,2%**

Table 1. Official results for the Språkdata-ML system

Conclusion

The existence of sense ambiguity (polysemy and homonymy) is one of the major problems affecting the usefulness of basic corpus exploration tools. In this respect, we regard sense disambiguation as a very important process and component when it is seen in the context of a wider and deeper text-processing architecture. In this paper we have described a simple feature-vector extraction approach to sense disambiguation that was utilized in a MBL software. We do not believe that we have fully exploited the capabilities of either the software or the way we can model the available resources. These issues will be investigated in the future, as well as the evaluation of the sense-tagger on an even larger scale.

References

Daelemans W., Zavrel J., van der Sloot K. and van den Bosch A. (1999). *TiMBL: Tilburg Memory Based Learner, version 2.0, Reference Guide*. ILK Technical Report 99-01, Paper available from: http://ilk.kub.nl/~ilk/papers/ilk9901.ps.gz.

Kokkinakis D. and Johansson Kokkinakis S. (1999). Sense Tagging at the Cycle-Level Using GLDB. *Nordiska Studier i Lexikografi*, vol. 27:146-167. Gellerstam M., Jóhannesson K., Ralph B. and Rogström L. (eds). Nordiska Föreningen för Lexikografi & Meijerbergs Institut för Svensk Etymologisk Forskning.

Kokkinakis D., Toporowska Gronostaj M. and Warmenius K. (2000). Annotating, Disambiguating & Automatically Extending the Coverage of the Swedish SIMPLE Lexicon. *Proceedings of the 2nd Languages Resources and Evaluation Conference (LREC)*, vol. III:1397-1404. Athens, Hellas.

Mitchell T. M. (1997). *Machine Learning*. McGraw-Hill Series on Computer Science.