Memory-based learning of word translation

Maria Holmqvist
Department of Computer and Information Science
Linköpings universitet
marho@ida.liu.se

Abstract

A basic task in machine translation is to choose the right translation for source words with several possible translations in the target language. In this paper we treat word translation as a word sense disambiguation problem and train memory-based classifiers on words with alternative translations. The training data was automatically labeled with the corresponding translations by word-aligning a parallel corpus. Results show that many words were translated with accuracy above the baseline.

1 Introduction

A problem in machine translation (MT) is choosing the right translation of a word or phrase with several equivalents in the target language. This problem of choosing the right translation of a source word in context is closely related to the well-researched problem of word sense disambiguation (WSD) which is the task of identifying the correct sense of a semantically ambiguous word in context.

In this paper we describe an experiment with machine learning of word translation from English to Swedish within a corpus-based approach to machine translation: direct (word-based) translation (Ahrenberg and Holmqvist, 2004). In the experiment we used lexical and grammatical correspondence information derived from parallel texts and a memory-based learning algorithm (Daelemans, 1999) to learn which translation should be used in a certain context during automatic translation.

Memory-based learning and similar machine learning techniques[1] have been successfully applied to the related problem of word sense disambiguation (Mihalcea, 2002; Ng and Lee, 1996; Hoste et al., 2002). Several other machine learning approaches used for word sense disambiguation have had similar success with the word translation task, including support-vector machines (Murata et al., 2001) and maximum entropy (Vickrey et al., 2005). The benefits of using a dedicated WSD/machine learning algorithm over a typical MT method like statistical MT, is that the WSD algorithm can take advantage of many types of contextual information not just a window of surrounding word forms. Recent attempts to combine the benefits of WSD and statistical methods include Carpuat and Wu (2005) and Vickrey et al. (2005).

2 Background

2.1 Word translation variation

Although the word translation task is a challenge for most MT systems, it is especially challenging for corpus-based methods that rely on information from real-world translation examples. Translations made by professional translators contain more variation in syntax and word choice than what is strictly necessary to produce fluent and accurate translations. This variation is one of the challenges when we try to learn word translation directly from parallel corpora.

The following translation ambiguities from English–Swedish corpus data illustrate the range of variation in word translation:

chair	*ordförande* (person)
chair	*stol* (furniture)

[1] E.g., exemplar-based or instance-based learning.

select	*välj* (choose)
select	*markera* (a choice on the file menu)
You can	*du* kan (you can)
It tells *you*	Det beskriver *NULL* (it describes)

Only the first example, *chair*, is a pure WSD problem dealing with sense. For the other words it is less obvious why a translation is used instead of another. In the last example, the deletion of *you* is a result of a change of the sentence from active to passive mode. Perhaps a translation system will not have to handle all of these translation ambiguities. However, there is also a real possibility that this variation in the corpus reflects the translator's knowledge of the subject domain, and language specific text norms within this domain. For translation scholars, extracting such knowledge from a translation can be as rewarding as extracting knowledge about lexical and structural correspondences (Merkel et al., 2003).

2.2 Memory-based learning

In our word translation experiment we used memory-based learning to train classifiers (Daelemans, 1999). A memory based classifier avoids overgeneralization by storing all training examples as feature vectors in memory without pruning exceptional instances. At run time a new instance is compared to the stored examples and gets a classification according to the closest match (nearest neighbors) in the database. All learning was done using the TiMBL software package[2] and the associated Paramsearch tool was used to optimize the parameter settings for the learning algorithm (Daelemans et al., 2004).

3 Learning word translation: An experiment

We conducted an experiment to find out to what extent correct translation of ambiguous words from the source context can be predicted using memory-based learning (Daelemans, 1999). We define the learning task as follows: for a lemma *a* and its context in the source language *S*, find the correct translation (lemma) in the target language *T*. The source context was represented by automatically selected contextual features based on a set of features used successfully in word sense disambiguation (Mihalcea, 2002).

3.1 The parallel corpus

The translation data was extracted from parallel texts from the English–Swedish translation of a database software manual. Both source and target texts are linguistically annotated with a dependency parser[3] that provides each word with lemma, part-of-speech, morphology and dependency relations (Tapanainen and Järvinen, 1997). The corpus is relatively small, consisting of 5382 aligned sentence pairs. To extract lexical correspondences (words and their translations) the corpus was word aligned with a combination of manual and automatic alignment. The first 1000 sentences were word aligned manually and the remaining 4382 sentences were aligned using the automatic word alignment tool I*Trix (Merkel et al., 2003).

3.2 Word types

We trained translation classifiers on 34 ambiguous word types that were selected from the corpus based on two criteria:

1. Their frequency in the manually aligned part of the training corpus.
2. Word alignment quality.

Instances of each word were extracted from the entire corpus and randomly assigned to train and test data containing 2/3 and 1/3 of the instances respectively. Afterwards, word alignments of instances in the test data were manually corrected in order to obtain gold standard translations for the evaluation of our classifiers. The manual correction showed that the alignment accuracy of the 34 selected words ranged from 78% to 100%. The number of target alternatives ranges from 2 to 11 and the most frequent target baseline ranged from 27% to 98%. Table 1 shows a sample of selected words.

Word	Targets
change (verb)	använda, byta, ändra
in (prep)	på, med, i

Table 1. Two of the selected words.

3.3 Filtering noisy training data

Since training data was extracted from automatic word alignment of a parallel corpus the classifications (targets) in training data contained noise. We therefore decided to try and filter the data by

[2] Tilburg Memory-based learner, http://ilk.uvt.nl/timbl/.

[3] Connexor Machinese Syntax, http://www.conexor.fi/.

removing instances with targets that did not occur in the manually aligned portion of the data.

The effects of filtering noisy data were investigated by comparing classifiers based on the original training data with classifiers based on the filtered data.

3.4 Feature selection

Careful selection of features and tuning of algorithm parameters are vital for machine learning performance. Tuning must be performed individually for each word classifier and in this experiment we used the feature selection procedure *forward selection* to optimize the features for each word. Starting with an empty set of features and another set of candidate features, each feature was tested using "leave one out"-testing on the training data. The candidate feature which improved the classification the most was selected and added to the feature set. This process of trying out candidate features was repeated until there was no more improvement in classification accuracy.

The set of candidate features in Table 2 was inspired by the ones used by Mihalcea (2002) for the WSD task, but we included features that use dependency information, such as properties of the head and daughter word. If a feature was not present in the context it was replaced with a default value, MISS. The features Col (collocations) and SK (sense specific keywords) are binary features that represent whether the collocation/keyword is present in the current sentence context or not. Following Ng and Lee (1996), the keywords and collocations for each word type are those that (1) occur at least 5 times with a target, and (2) have a conditional probability above 0.8, where the conditional probability is the number of times a keyword/collocation occurs with a word *w* with target word *t* divided with all occurrences of word *w*. For the noun *data* this produced the following set of collocations:

```
data access, data sources, data access page, offline data, data in a,
data in, data from
```

and keywords:

```
sources, report, fields, list,
form, PivotTable
```

3.5 TiMBL parameter settings

The TiMBL parameter settings were also individually set for each word learning task. Ideally, parameter and feature optimization should be interleaved and exhaustive. As a decent compromise, we ran the Paramsearch utility to optimize the parameter settings (feature weighting, number of k-nearest neighbors and distance metric) each time a new feature was selected from the set of candidate features.

Dependency features						
	Form	Base	PoS	Morf.	Dep. rel	
Current word	x		x	x	x	
Head	x	x	x	x	x	
Right daughter	x	x	x	x	x	
Left daughter	x	x	x	x	x	
Head of NP		x		x		
Surface features						
	N	V	NE	Prep	Pron	Det
Before	x	x	x	x	x	x
After	x	x	x	x	x	
Other features						
CF	Words and PoS in window size -1, +2					
Col	Collocations (Ng and Lee, 1996) (max. 5)					
SK	Sense specific keywords (max. 5)					

Table 2. Candidate features.

3.6 Results

Table 3 shows the average results of training and testing on all word types using filtered and unfiltered training data. The results are compared against a baseline of applying the most frequent translation found in training data. Results show that on average, the memory-based classifiers did better on the word translation task than the simple baseline. However, for both types of training data only about 60% of the words were more accurate than the baseline. By filtering the noisy training data we also achieved better results. However, this improvement was rather modest and was not consistent over all word types.

	Original	**Filtered**	**Baseline**
Accuracy	67.2%	70.7%	63.3%

Table 3. Average accuracy for all word types with original and filtered data.

It is also clear from comparing the memory-based classifiers to the baseline in Figure 1, that the classifiers have considerable difficulties competing with high baseline accuracy. For the majority of words with a baseline accuracy over

60% the memory-based method achieved an accuracy equal to or less than baseline.

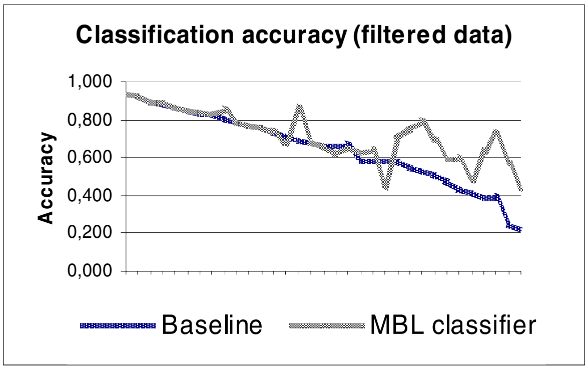

Figure 1. Classification accuracy compared to baseline accuracy (most frequent target).

The best contextual features for classification were automatically selected for each word type. For example, the classifier for preposition *on* selected features RightDaughterLemma and NounBefore to compare instances. The classifier for the word type *you* trained on the original data set used the Keyword feature. Here the presence or absence of the keywords: *move*, *drag*, *ID*, *indexes*, and *spreadsheet* turned out to be useful contextual features for deciding on the correct translation.

Interestingly, many of the selected features were features that we derived from the relations in the dependency parsetree. Table 4 presents the surrounding words used as contextual features for the nouns in the experiment.

Nouns	**Features**
argument	CurrentNode
custom	Head
data	LeftDaughter VerbBefore
item	NamedEntityAfter PronounAfter
view	Head LeftDaughter NamedEntityBefore

Table 4. Contextual features selected for nouns.

4 Conclusion

We have carried out an experiment with memory-based learning of word translation to see if we can train useful classifiers for this task, despite the noisy data produced by automatic word alignment. Results show that our memory-based classifier in many cases will be more accurate in predicting translations than a baseline classifier, especially on words with a baseline accuracy of less than 60%. We also showed that dependency type features were found to be useful contextual cues for deciding the correct translations of words.

References

Lars Ahrenberg and Maria Holmqvist. 2004. Back to the future? The case for English–Swedish direct machine translation. In *Proc. of RASMAT‘2004*.

Marine Carpuat and Dekai Wu. 2005. Word sense disambiguation vs. statistical machine translation. In *Proc. of ACL’2005*, pp. 387–394.

Walter Daelemans. 1999. Memory-based language processing. *Journal for Experimental and Theoretical Artificial Intelligence,* 11(3), pp. 287–467.

Walter Daelemans, Jakub Zavrel, Ko Van der Sloot, and Antal Van den Bosch. 2004. *TiMBL: Tilburg Memory Based Learner, version 5.1, Reference Guide.*

Vèronique Hoste, Iris Hendrickx, Walter Daelemans, and Antal van den Bosch. 2002. Parameter optimization for machine learning of word sense disambiguation. *Natural Language Engineering,* 8(4), pp. 311–325,

Magnus Merkel, Michael Petterstedt, and Lars Ahrenberg. 2003. Interactive Word Alignment for Corpus Linguistics. In *Proc. from Corpus Linguistics 2003.*

Rada Mihalcea. 2002. Instance based learning with automatic feature selection applied to word sense disambiguation. In *Proc. of COLING 2002*, pp. 1–10.

Masaki Murata, Kiyotaka Uchimoto, Qing Ma, and Hitoshi Isahara. 2001. Using a support-vector machine for Japanese-to-English translation of tense, aspect, and modality. In *Proc. of ACL’2001 Workshop on Data-Driven Methods in Machine Translation,* pp. 1–8.

Hwee Tou Ng and Hian Beng Lee. 1996. Integrating-multiple knowledge sources to disambiguate word-sense: An exemplar-based approach. In *Proc. of the 34th Annual Meeting of the ACL*, pp. 40–47.

Pasi Tapanainen and Timo Järvinen. 1997. A non-projective dependency parser. In *Proc. of ANLP’1997*, pp. 64–71.

David Vickrey, Luke Biewald, Marc Teyssier, and Daphne Koller. 2005. Word-sense disambiguation for machine translation. In *Proc. of EMNLP’2005*, pp. 771–778.