HLTDI: CL-WSD Using Markov Random Fields for SemEval-2013 Task 10

Alex Rudnick, Can Liu and Michael Gasser
Indiana University, School of Informatics and Computing
{alexr,liucan,gasser}@indiana.edu

Abstract

We present our entries for the SemEval-2013 cross-language word-sense disambiguation task (Lefever and Hoste, 2013). We submitted three systems based on classifiers trained on local context features, with some elaborations. Our three systems, in increasing order of complexity, were: maximum entropy classifiers trained to predict the desired target-language phrase using only monolingual features (we called this system L1); similar classifiers, but with the desired target-language phrase for the other four languages as features (L2); and lastly, networks of five classifiers, over which we do loopy belief propagation to solve the classification tasks jointly (MRF).

1 Introduction

In the cross-language word-sense disambiguation (CL-WSD) task, given an instance of an ambiguous word used in a context, we want to predict the appropriate translation into some target language. This setting for WSD has an immediate application in machine translation, since many words have multiple possible translations. Framing the resolution of lexical ambiguities as an explicit classification task has a long history, and was considered in early SMT work at IBM (Brown et al., 1991). More recently, Carpuat and Wu have shown how to use CL-WSD techniques to improve modern phrase-based SMT systems (Carpuat and Wu, 2007), even though the language model and phrase-tables of these systems mitigate the problem of lexical ambiguities somewhat.

In the SemEval-2013 CL-WSD shared task (Lefever and Hoste, 2013), entrants are asked to build a system that can provide translations for twenty ambiguous English nouns, given appropriate contexts – here the particular usage of the ambiguous noun is called the *target* word. The five target languages of the shared task are Spanish, Dutch, German, Italian and French. In the evaluation, for each of the twenty ambiguous nouns, systems are to provide translations for the target word in each of fifty sentences or short passages. The translations of each English word may be single words or short phrases in the target language, but in either case, they should be lemmatized.

Following the work of Lefever and Hoste (2011), we wanted to make use of multiple bitext corpora for the CL-WSD task. ParaSense, the system of Lefever and Hoste, takes into account evidence from all of the available parallel corpora. Let S be the set of five target languages and t be the particular target language of interest at the moment; ParaSense creates bag-of-words features from the translations of the target sentence into the languages $S - \{t\}$. Given corpora that are parallel over many languages, this is straightforward at training time. However, at testing time it requires a complete MT system for each of the four other languages, which is computationally prohibitive. Thus in our work, we learn from several parallel corpora but require neither a locally running MT system nor access to an online translation API.

We presented three systems in this shared task, all of which were variations on the theme of a maximum entropy classifier for each ambiguous noun, trained on local context features similar to those used in previous work and familiar from the WSD literature. The first system, L1 ("layer one"), uses maximum entropy classifiers trained on local context features. The second system, L2 ("layer two"), is the same as the L1 system, with the addition of the correct translations into the other target languages as features, which at testing time are predicted with L1 classifiers. The third system, MRF ("Markov random field") uses a network of interacting classifiers to solve the classification problem for all five target languages jointly. Our three systems are all trained from the same data, which we extracted from the Europarl Intersection corpus provided by the shared task organizers.

At the time of the evaluation, our simplest system had the top results in the shared task for the out-of-five evaluation for three languages (Spanish, German, and Italian). However, after the evaluation deadline, we fixed a simple bug in our MRF code, and the MRF system then achieved even better results for the *oof* evaluation. For the *best* evaluation, our two more sophisticated systems posted better results than the L1 version. All of our systems beat the "most-frequent sense" baseline in every case.

In the following sections, we will describe our three systems[1], our training data extraction process, the results on the shared task, and conclusions and future work.

2 L1

The "layer one" classifier, L1, is a maximum entropy classifier that uses only monolingual features from English. Although this shared task is described as unsupervised, the L1 classifiers are trained with supervised learning on instances that we extract programmatically from the Europarl Intersection corpus; we describe the preprocessing and training data extraction in Section 5.

Having extracted the relevant training sentences from the aligned bitext for each of the five language pairs, we created training instances with local context features commonly used in WSD systems. These are described in Figure 1. Each instance is assigned the lemma of the translation that was extracted from the training sentence as its label.

- target word features
  - literal word form
  - POS tag
  - lemma
- window unigram features (within 3 words)
  - word form
  - POS tag
  - word with POS tag
  - word lemma
- window bigram features (within 5 words)
  - bigrams
  - bigrams with POS tags

Figure 1: Features used in our classifiers

We trained one L1 classifier for each target language and each word of interest, resulting in $20*5 = 100$ classifiers. Classifiers were trained with the MEGA Model optimization package [2] and its corresponding NLTK interface (Bird et al., 2009). Upon training, we cache these classifiers with Python pickles, both to speed up L1 experiments and also because they are used as components of the other models.

We combined the word tokens with their tags in some features so that the classifier would not treat them independently, since maximum entropy classifiers learn a single weight for each feature. Particularly, the "POS tag" feature is distinct from the "word with tag" feature; for the tagged word "house/NN", the "POS tag" feature would be NN, and the "word with tag" feature is $house\_NN$.

3 L2

The "layer two" classifier, L2, is an extension to the L1 approach, with the addition of multilingual features. Particularly, L2 makes use of the translations of the target word into the four target languages other than the one we are currently trying to predict. At training time, since we have the translations of each of the English sentences into the other target languages, the appropriate features are extracted from the corresponding sentences in those languages. This is the same as the process by which labels are given to training instances, described in Section 5. At testing time, since translations of the

[1] Source is available at
http://github.iu.edu/alexr/semeval2013

[2] http://www.umiacs.umd.edu/~hal/megam/

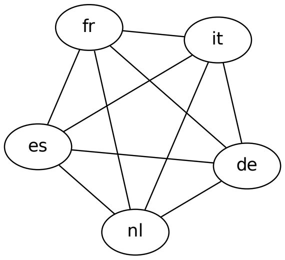

Figure 2: The network structure used in the MRF system: a complete graph with five nodes where each node represents a variable for the translation into a target language

test sentences are not given, we estimate the translations for the target word in the four other languages using the cached L1 classifiers.

Lefever and Hoste (2011) used the Google Translate API to translate the source English sentences into the four other languages, and extracted bag-of-words features from these complete sentences. The L2 classifiers make use of a similar intuition, but they do not rely on a complete MT system or an available online MT API; we only include the translations of the specific target word as features.

4 MRF

Our MRF model builds a Markov network (often called a "Markov random field") of L1 classifiers in an effort to find the best translation into all five target languages jointly. This network has nodes that correspond to the distributions produced by the L1 classifiers, given an input sentence, and edges with pairwise potentials that are derived from the joint probabilities of target-language labels occurring together in the training data. Thus the task of finding the optimal translations into five languages jointly is framed as a MAP (Maximum A Posteriori) inference problem, where we try to maximize the joint probability $P(w_{fr}, w_{es}, w_{it}, w_{de}, w_{nl})$, given the evidence of the features extracted from the source-language sentence. The inference process is performed using loopy belief propagation (Murphy et al., 1999), which is an approximate but tractable inference algorithm that, while it gives no guarantees, often produces good solutions in practice.

The intuition behind using a Markov network for this task is that, since we must make five decisions for each source-language sentence, we should make use of the correlations between the target-language words. Correlations might occur in practice due to cognates – the languages in the shared task are fairly closely related – or they may simply reflect ambiguities in the source language that are resolved in two target languages.

So by building a Markov network in which all of the classifiers can communicate (see Figure 2), we allow nodes to influence the translation decisions of their neighbors, but only proportionally to the correlation between the translations that we observe in the two languages.

We frame the MAP inference task as a minimization problem; we want to find an assignment that minimizes the sum of all of our penalty functions, which we will describe next. First, we have a unary function from each of the five L1 classifiers, which correspond to nodes in the network. These functions each assign a penalty to each possible label for the target word in the corresponding language; that penalty is simply the negative log of the probability of the label, as estimated by the classifier.

Formally, a unary potential ϕ_i, for some fixed set of features f and a particular language i, is a function from a label l to some positive penalty value.

$$\phi_i(l) = -logP(L_i = l|F = f)$$

Secondly, for each unordered pair of classifiers (i, j) (*i.e.*, each edge in the graph) there is a pairwise potential function $\phi_{(i,j)}$ that assigns a penalty to any assignment of that pair of variables.

$$\phi_{(i,j)}(l_i, l_j) = -logP(L_i = l_i, L_j = l_j)$$

Here by $P(L_i = l_i, L_j = l_j)$, we mean the probability that, for a fixed ambiguous input word, language i takes the label l_i and language j takes the label l_j. These joint probabilities are estimated from the training data; we count the number of times each pair of labels l_i and l_j co-occurs in the training sentences and divide, with smoothing to avoid zero probabilities and thus infinite penalties.

When it comes time to choose translations, we want to find a complete assignment to the five variables that minimizes the sum of all of the penalties assigned by the ϕ functions. As mentioned earlier, we do this via loopy belief propagation, using the formulation for pairwise Markov networks that passes messages directly between the nodes rather than first constructing a cluster graph (Koller and Friedman, 2009, §11.3.5.1).

As we are trying to compute the minimum-penalty assignment to the five variables, we use the *min-sum* version of loopy belief propagation. The messages are mappings from the possible values that the recipient node could take to penalty values.

At each time step, every node passes to each of its neighbors a message of the following form:

$$\delta^t_{i\rightarrow j}(L_j) = \min_{l_i \in L_i} \Big[\phi_i(l_i) + \phi_{(i,j)}(l_i, l_j) + \sum_{k \in S - \{i,j\}} \delta^{t-1}_{k \rightarrow i}(l_i) \Big]$$

By this expression, we mean that the message from node i to node j at time t is a function from possible labels for node j to scalar penalty values. Each penalty value is determined by minimizing over the possible labels for node i, such that we find the label l_i that minimizes sum of the unary cost for that label, the binary cost for l_i and l_j taken jointly, and all of the penalties in the messages that node i received at the previous time step, except for the one from node j.

Intuitively, these messages inform a given neighbor about the estimate, from the perspective of the sending node and what it has heard from its other neighbors, of the minimum penalty that would be incurred if the recipient node were to take a given label. As a concrete example, when the *nl* node sends a message to the *fr* node at time step 10, this message is a table mapping from all possible French translations of the current target word to their associated penalty values. The message depends on three things: the function ϕ_{nl} (itself dependent on the probability distribution output by the L1 classifier), the binary potential function $\phi_{(nl,fr)}$, and the messages from *es*, *it* and *de* from time step 9. Note that the binary potential functions are symmetric because they are derived from joint probabilities.

Loopy belief propagation is an approximate inference algorithm, and it is neither guaranteed to find a globally optimal solution, nor even to converge at all, but it does often find good solutions in practice. We run it for twenty iterations, which empirically works well. After the message-passing iterations, each node chooses the value that minimizes the sum of the penalties from messages and from its own unary potential function. To avoid accumulating very large penalties, we normalize the outgoing messages at each time step and give a larger weight to the unary potential functions. These normalization and weighting parameters were set by hand, but seem to work well in practice.

5 Training Data Extraction

For simplicity and comparability with previous work, we worked with the Europarl Intersection corpus provided by the task organizers. Europarl (Koehn, 2005) is a parallel corpus of proceedings of the European Parliament, currently available in 21 European languages, although not every sentence is translated into every language. The Europarl Intersection is the intersection of the sentences from Europarl that are available in English and all five of the target languages for the task.

In order to produce the training data for the classifiers, we first tokenized the text for all six languages with the default NLTK tokenizer and tagged the English text with the Stanford Tagger (Toutanova et al., 2003). We aligned the untagged English with each of the target languages using the Berkeley Aligner (DeNero and Klein, 2007) to get one-to-many alignments from English to target-language words, since the target-language labels may be multi-word phrases. We used nearly the default settings for Berkeley Aligner, except that we ran 20 iterations each of IBM Model 1 and HMM alignment.

We used TreeTagger (Schmid, 1995) to lemmatize the text. At first this caused some confusion in our pipeline, as TreeTagger by default re-tokenizes input text and tries to recognize multi-word expressions. Both of these, while sensible behaviors, were unexpected, and resulted in a surprising number of tokens in the TreeTagger output. Once we turned off these behaviors, TreeTagger provided useful lemmas for all of the languages.

Given the tokenized and aligned sentences, with their part-of-speech tags and lemmas, we used a number of heuristics to extract the appropriate target-language labels for each English-language input sentence. For each target word, we extracted a sense inventory V_i from the gold standard answers from the 2010 iteration of this task (Lefever and Hoste, 2009). Then, for each English sentence that contains one of the target words used as a noun, we examine the alignments to determine whether that word is aligned with a sense present in V_i , or whether the words aligned to that noun are a subsequence of such a sense. The same check is performed both on the lemmatized and unlemmatized versions of the target-language sentence. If we do find a match, then that sense from the gold standard V_i is taken to be the label for this sentence. While a gold standard sense inventory will clearly not be present for general translation systems, there will be some vocabulary of possible translations for each word, taken from a bilingual dictionary or the phrase table in a phrase-based SMT system.

If a label from V_i is not found with the alignments, but some other word or phrase is aligned with the ambiguous noun, then we trust the output of the aligner, and the lemmatized version of this target-language phrase is assigned as the label for this sentence. In this case we used some heuristic functions to remove stray punctuation and attached articles (such as *d'* from French or *nell'* from Italian) that were often left appended to the tokens by the default NLTK English tokenizer.

We dropped all of the training instances with labels that only occurred once, considering them likely alignment errors or other noise.

6 Results

There were two settings for the evaluation, *best* and *oof*. In either case, systems may present multiple possible answers for a given translation, although in the *best* setting, the first answer is given more weight in the evaluation, and the scoring encourages only returning the top answer. In the *oof* setting, systems are asked to return the top-five most likely translations. In both settings, the answers are compared against translations provided by several human annotators for each test sentence, who provided a number of possible target-language translations in lemmatized form, and more points are given for matching the more popular translations given by the annotators. In the "mode" variant of scoring, only the one most common answer for a given test sentence is considered valid. For a complete explanation of the evaluation and its scoring, please see the shared task description (Lefever and Hoste, 2013).

The scores for our systems[3] are reported in Figure 3. In all of the settings, our systems posted some of the top results among entrants in the shared task, achieving the best scores for some evaluations and some languages. For every setting and language, our systems beat the most-frequent sense baseline, and our best results usually came from either the L2 or MRF system, which suggests that there is some benefit in using multilingual information from the parallel corpora, even without translating the whole source sentence.

For the *best* evaluation, considering only the mode gold-standard answers, our L2 system achieved the highest scores in the competition for Spanish and German. For the *oof* evaluation, our MRF system – with its post-competition bug fix – posted the best results for Spanish, German and Italian in both complete and mode variants. Also, curiously, our L1 system posted the best results in the competition for Dutch in the *oof* variant.

For the *best* evaluation, our results were lower than those posted by ParaSense, and in the standard *best* setting, they were also lower than those from the *c1lN* system (van Gompel and van den Bosch, 2013) and *adapt1* (Carpuat, 2013). This, combined with the relatively small difference between our simplest system and the more sophisticated ones, suggests that there are many improvements that could be made to our system; perhaps

[3] The *oof* scores for the MRF system reflect a small bug fix after the competition.

system	es	nl	de	it	fr
MFS	23.23	20.66	17.43	20.21	25.74
best	32.16	23.61	20.82	25.66	30.11
PS	31.72	25.29	24.54	28.15	31.21
L1	29.01	21.53	19.5	24.52	27.01
L2	28.49	**22.36**	**19.92**	23.94	**28.23**
MRF	**29.36**	21.61	19.76	**24.62**	27.46

(a) *best* evaluation results: precision

system	es	nl	de	it	fr
MFS	53.07	43.59	38.86	42.63	51.36
best	62.21	47.83	44.02	53.98	59.80
L1	61.69	46.55	43.66	53.57	57.76
L2	59.51	46.36	42.32	53.05	**58.20**
MRF	***62.21***	**46.63**	***44.02***	***53.98***	57.83

(b) *oof* evaluation results: precision

system	es	nl	de	it	fr
MFS	27.48	24.15	15.30	19.88	20.19
best	37.11	27.96	24.74	31.61	26.62
PS	40.26	30.29	25.48	30.11	26.33
L1	36.32	25.39	24.16	26.52	**21.24**
L2	***37.11***	25.34	***24.74***	**26.65**	21.07
MRF	36.57	**25.72**	24.01	26.26	**21.24**

(c) *best* evaluation results: mode precision

system	es	nl	de	it	fr
MFS	57.35	41.97	44.35	41.69	47.42
best	65.10	47.34	53.75	57.50	57.57
L1	64.65	***47.34***	53.50	56.61	51.96
L2	62.52	44.06	49.03	54.06	**53.57**
MRF	***65.10***	47.29	***53.75***	***57.50***	52.14

(d) *oof* evaluation results: mode precision

Figure 3: Task results for our systems. Scores in **bold** are the best result for that language and evaluation out of our systems, and those in ***bold italics*** are the best posted in the competition. For comparison, we also give scores for the most-frequent-sense baseline ("MFS"), ParaSense ("PS"), the system developed by Lefever and Hoste, and the best posted score for competing systems this year ("best").

we could integrate ideas from the other entries in the shared task this year.

7 Conclusions and future work

Our systems had a strong showing in the competition, always beating the MFS baseline, achieving the top score for three of the five languages in the *oof* evaluation, and for two languages in the *best* evaluation when considering the mode gold-standard answers. The systems that took into account evidence from multiple sources had better performance than the one using monolingual features: our top result in every language came from either the L2 or the MRF classifier for both evaluations. This suggests that it is possible to make use of the evidence in several parallel corpora in a CL-WSD task without translating every word in a source sentence into many target languages.

We expect that the L2 classifier could be improved by adding features derived from more classifiers and making use of information from many disparate sources. We would like to try adding classifiers trained on the other Europarl languages, as well as completely different corpora. The L2 classifier approach only requires that the first-layer classifiers make *some* prediction based on text in the source language. They need not be trained from the same source text, depend on the same features, or even output words as labels. In future work we will explore all of these variations. One could, for example, train a monolingual WSD system on a sense-tagged corpus and use this as an additional information source for an L2 classifier.

There remain a number of avenues that we would like to explore for the MRF system; thus far, we have used the joint probability of two labels to set the binary potentials. We would like to investigate other functions, especially ones that do not incur large penalties for rare labels, as the joint probability of two labels that often co-occur but are both rare will be low. Also, in the current system, the relative weights of the binary potentials and the unary potentials were set by hand, with a very small amount of empirical tuning. We could, in the future, tune the weights with a more principled optimization strategy, using a development set.

As with the L2 classifiers, it would be helpful in the future for the MRF system to not require many mutually parallel corpora for training – however, the current approach for estimating the edge potentials requires the use of bitext for each edge in the network. Perhaps these correlations could be estimated in a semi-supervised way, with high-confidence automatic labels being used to estimate the joint distribution over target-language phrases. We would also like to investigate approaches to jointly disambiguate many words in the same sentence, since lexical ambiguity is not just a problem for a few nouns.

Aside from improvements to the design of our CL-WSD system itself, we want to use it in a practical system for translating into under-resourced languages. We are now working on integrating this project with our rule-based MT system, L^3 (Gasser, 2012). We had experimented with a similar, though less sophisticated, CL-WSD system for Quechua (Rudnick, 2011), but in the future, L^3 with the integrated CL-WSD system should be capable of translating Spanish to Guarani, either as a standalone system, or as part of a computer-assisted translation tool.

References

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

Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1991. Word-Sense Disambiguation Using Statistical Methods. In *Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics*, pages 264–270.

Marine Carpuat and Dekai Wu. 2007. How Phrase Sense Disambiguation Outperforms Word Sense Disambiguation for Statistical Machine Translation. In *11th Conference on Theoretical and Methodological Issues in Machine Translation*.

Marine Carpuat. 2013. NRC: A Machine Translation Approach to Cross-Lingual Word Sense Disambiguation (SemEval-2013 Task 10). In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, Atlanta, USA.

John DeNero and Dan Klein. 2007. Tailoring Word Alignments to Syntactic Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 17–24, Prague, Czech Republic, June. Association for Computational Linguistics.

Michael Gasser. 2012. Toward a Rule-Based System for English-Amharic Translation. In *LREC-2012: SALTMIL-AfLaT Workshop on Language technology for normalisation of less-resourced languages*.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of The Tenth Machine Translation Summit*, Phuket, Thailand.

D. Koller and N. Friedman. 2009. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press.

Els Lefever and Véronique Hoste. 2009. SemEval-2010 Task 3: Cross-lingual Word Sense Disambiguation. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions (SEW-2009)*, pages 82–87, Boulder, Colorado, June. Association for Computational Linguistics.

Els Lefever and Véronique Hoste. 2013. SemEval-2013 Task 10: Cross-Lingual Word Sense Disambiguation. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, Atlanta, USA.

Els Lefever, Véronique Hoste, and Martine De Cock. 2011. ParaSense or How to Use Parallel Corpora for Word Sense Disambiguation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 317–322, Portland, Oregon, USA, June. Association for Computational Linguistics.

Kevin P. Murphy, Yair Weiss, and Michael I. Jordan. 1999. Loopy Belief Propagation for Approximate Inference: An Empirical Study. In *UAI '99: Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, Stockholm, Sweden.

Alex Rudnick. 2011. Towards Cross-Language Word Sense Disambiguation for Quechua. In *Proceedings of the Second Student Research Workshop associated with RANLP 2011*, pages 133–138, Hissar, Bulgaria, September. RANLP 2011 Organising Committee.

Helmut Schmid. 1995. Improvements In Part-of-Speech Tagging With an Application To German. In *Proceedings of the ACL SIGDAT-Workshop*, pages 47–50.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *PROCEEDINGS OF HLT-NAACL*, pages 252–259.

Maarten van Gompel and Antal van den Bosch. 2013. WSD2: Parameter optimisation for Memory-based Cross-Lingual Word-Sense Disambiguation. In *Proceedings of the 7th International Workshop on Semantic Evaluation (SemEval 2013)*, Atlanta, USA.