# Multilingual semantic parsing with a pipeline of linear classifiers

**Oscar Täckström**
Swedish Institute of Computer Science
SE-16429, Kista, Sweden
oscar@sics.se

## Abstract

I describe a fast multilingual parser for semantic dependencies. The parser is implemented as a pipeline of linear classifiers trained with support vector machines. I use only first order features, and no pair-wise feature combinations in order to reduce training and prediction times. Hyper-parameters are carefully tuned for each language and sub-problem.

The system is evaluated on seven different languages: Catalan, Chinese, Czech, English, German, Japanese and Spanish. An analysis of learning rates and of the reliance on syntactic parsing quality shows that only modest improvements could be expected for most languages given more training data; Better syntactic parsing quality, on the other hand, could greatly improve the results. Individual tuning of hyper-parameters is crucial for obtaining good semantic parsing quality.

## 1 Introduction

This paper presents my submission for the semantic parsing track of the CoNLL 2009 shared task on syntactic and semantic dependencies in multiple languages (Hajič et al., 2009). The submitted parser is simpler than the submission in which I participated at the CoNLL 2008 shared task on joint learning of syntactic and semantic dependencies (Surdeanu et al., 2008), in which we used a more complex committee based approach to both syntax and semantics (Samuelsson et al., 2008). Results are on par with our previous system, while the parser is orders of magnitude faster both at training and prediction time and is able to process natural language text in Catalan, Chinese, Czech, English, German, Japanese and Spanish. The parser depends on the input to be annotated with part-of-speech tags and syntactic dependencies.

## 2 Semantic parser

The semantic parser is implemented as a pipeline of linear classifiers and a greedy constraint satisfaction post-processing step. The implementation is very similar to the best performing subsystem of the committee based system in Samuelsson et al. (2008).

Parsing consists of four steps: *predicate sense disambiguation*, *argument identification*, *argument classification* and *predicate frame constraint satisfaction*. The first three steps are implemented using linear classifiers, along with heuristic filtering techniques. Classifiers are trained using the support vector machine implementation provided by the LIBLINEAR software (Fan et al., 2008). MALLET is used as a framework for the system (McCallum, 2002).

For each classifier, the $c$-parameter of the SVM is optimised by a one dimensional grid search using threefold cross validation on the training set. For the identification step, the $c$-parameter is optimised with respect to $F_1$-score of the positive class, while for sense disambiguation and argument labelling the optimisation is with respect to accuracy. The regions to search were identified by initial runs on the development data. Optimising these parameters for each classification problem individually proved to be crucial for obtaining good results.

### 2.1 Predicate sense disambiguation

Since disambiguation of predicate sense is a multiclass problem, I train the classifiers using the method of Crammer and Singer (2002), using the implementation provided by LIBLINEAR. Sense labels do not generalise over predicate lemmas, so one classifier is trained for each lemma occurring in the training data. Rare predicates are given the most common sense of the predicate. Predicates occurring less than

*Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL): Shared Task*, pages 103–108,
Boulder, Colorado, June 2009. ©2009 Association for Computational Linguistics

7 times in the training data were heuristically determined to be considered rare. Predicates with unseen lemmas are labelled with the most common sense tag in the training data.

### 2.1.1 Feature templates

The following feature templates are used for predicate sense disambiguation:

PredicateWord
Predicate[PoS/Feats]
PredicateWindowBagLemmas
PredicateWindowPosition[PoS/Feats]
GovernorRelation
Governor[Word/Lemma]
Governor[PoS/Feats]
DependentRelation
Dependent[Word/Lemma]
Dependent[PoS/Feats]
DependentSubCat.

The *window feature templates extract features from the two preceding and the two following tokens around the predicate, with respect to the linear ordering of the tokens. The *Feats templates are based on information in the pfeats input column for the languages where this information is provided.

## 2.2 Argument identification and labelling

In line with most previous pipelined systems, identification and labelling of arguments are performed as two separate steps. The classifiers in the identification step are trained with the standard $L_2$-loss SVM formulation, while the classifiers in the labelling step are trained using the method of Crammer and Singer.

In order to reduce the number of candidate arguments in the identification step, I apply the filtering technique of Xue and Palmer (2004), trivially adopted to the dependency syntax formalism. Further, a filtering heuristic is applied in which argument candidates with rare predicate / argument part-of-speech combinations are removed; *rare* meaning that the argument candidate is actually an argument in less than 0.05% of the occurrences of the pair. These heuristics greatly reduce the number of instances in the argument identification step and improve performance by reducing noise from the training data.

Separate classifiers are trained for verbal predicates and for nominal predicates, both in order to save computational resources and because the frame structures do not generalise between verbal and nominal predicates. For Czech, in order to reduce training time I split the argument identification problem into three sub-problems: *verbs*, *nouns* and *others*, based on the part-of-speech of the predicate. In hindsight, after solving a file encoding related bug which affected the separability of the Czech data set, a split into verbal and nominal predicates would have sufficed. Unfortunately I was not able to rerun the Czech experiments on time.

### 2.2.1 Feature templates

The following feature templates are used both for argument identification and argument labelling:

PredicateLemmaSense
Predicate[PoS/Feats]
Position
Argument[PoS/Feats]
Argument[Word/Lemma]
ArgumentWindowPositionLemma
ArgumentWindowPosition[PoS/Feats]
LeftSiblingWord
LeftSibling[PoS/Feats]
RightSiblingWord
RightSibling[PoS/Feats]
LeftDependentWord
RightDependent[PoS/Feats]
RelationPath
TrigramRelationPath
GovernorRelation
GovernorLemma
Governor[PoS/Feats]

Most of these features, introduced by Gildea and Jurafsky (2002), belong to the folklore by now. The TrigramRelationPath is a ”soft” version of the RelationPath template, which treats the relation path as a bag of triplets of directional labelled dependency relations. Initial experiments suggested that this feature slightly improves performance, by overcoming local syntactic parse errors and data sparseness in the case of small training sets.

### 2.2.2 Predicate frame constraints

Following Johansson and Nugues (2008) I impose the Core argument consistency and Con-

TINUATION CONSISTENCY constraints on the generated semantic frames. In the cited work, these constraints are used to filter the candidate frames for a re-ranker. I instead perform a greedy search in which only the core argument with the highest score is kept when the former constraint is violated. The latter constraint is enforced by simply dropping any continuation argument lacking its corresponding core argument. Initial experiments on the development data indicates that these simple heuristics slightly improves semantic parsing quality measured with labelled $F_1$-score. It is possible that the improvement could be greater by using $L_2$-regularised logistic regression scores instead of the SVM scores, since the latter can not be interpreted as probabilities. However, logistic regression performed consistently worse than the SVM formulation of Crammer and Singer in the argument labelling step.

### 2.2.3 Handling of multi-function arguments

In Czech and Japanese an argument can have multiple relations to the same predicate, i.e. the semantic structure needs sometimes be represented by a multi-graph. I chose the simplest possible solution and treat these structures as ordinary graphs with complex labels. This solution is motivated by the fact that the palette of multi-function arguments is small, and that the multiple functions mostly are highly interdependent, such as in the ACT|PAT complex which is the most common in Czech.

## 3 Results

The semantic parser was evaluated on in-domain data for Catalan, Chinese, Czech, English, German, Japanese and Spanish, and on out-of-domain data for Czech, English and German. The respective data sets are described in Taulé et al. (2008), Palmer and Xue (2009), Hajič et al. (2006), Surdeanu et al. (2008), Burchardt et al. (2006) and Kawahara et al. (2002).

My official submission scores are given in table 1, together with post submission labelled and unlabelled $F_1$-scores. The official submissions were affected by bugs related to file encoding and hyperparameter search. After resolving these bugs, I obtained an improvement of mean $F_1$-score of almost 10 absolute points compared to the official scores.

| | Lab $F_1$ | Lab $F_1$ | Unlab $F_1$ |
|---|---|---|---|
| Catalan | 57.11 | 67.14 | 93.31 |
| Chinese | 63.41 | 74.14 | 82.57 |
| Czech | 71.05 | 78.29 | 89.20 |
| English | 67.64 | 78.93 | 88.70 |
| German | 53.42 | 62.98 | 89.64 |
| Japanese | 54.74 | 61.44 | 66.01 |
| Spanish | 61.51 | 69.93 | 93.54 |
| Mean | 61.27 | 70.41 | 86.14 |
| Czech$^\dagger$ | 71.59 | 78.77 | 87.13 |
| English$^\dagger$ | 59.82 | 68.96 | 86.23 |
| German$^\dagger$ | 50.43 | 47.81 | 79.52 |
| Mean$^\dagger$ | 60.61 | 65.18 | 84.29 |

Table 1: Semantic labelled and unlabelled $F_1$-scores for each language and domain. Left column: official labelled $F_1$-score. Middle column: post submission labelled $F_1$-score. Right column: post submission unlabelled $F_1$-score. $^\dagger$ indicates out-of-domain test data.

Clearly, there is a large difference in performance for the different languages and domains. As could be expected the parser performs much better for the languages for which a large training set is provided. However, as discussed in the next section, simply adding more training data does not seem to solve the problem.

Comparing unlabelled $F_1$-scores with labelled $F_1$-scores, it seems that argument identification and labelling errors contribute almost equally to the total errors for Chinese, Czech and English. For Catalan, Spanish and German argument identification scores are high, while labelling scores are in the lower range. Japanese stands out with exceptionally low identification scores. Given that the quality of the predicted syntactic parsing was higher for Japanese than for any other language, the bottleneck when performing semantic parsing seems to be the limited expressivity of the Japanese syntactic dependency annotation scheme.

Interestingly, for Czech, the result on the out-of-domain data set is better than the result on the in-domain data set, even though the unlabelled result is slightly worse. For English, on the other hand the performance drop is in the order of 10 absolute labelled $F_1$ points, while the drop in unlabelled $F_1$-score is comparably small. The result on German out-of-domain data seems to be an outlier, with post-submission results even worse than the official submission results. I suspect that this is due to a bug.

| | 10% | 25% | 50% | 75% | 100% |
|---|---|---|---|---|---|
| Catalan | 54.86 | 60.52 | 65.22 | 66.35 | 67.14 |
| Chinese | 72.93 | 73.40 | 73.77 | 74.08 | 74.14 |
| Czech | 75.42 | 76.90 | 77.69 | 78.00 | 78.29 |
| English | 75.75 | 77.56 | 78.37 | 78.71 | 78.93 |
| German | 47.77 | 54.74 | 58.94 | 61.02 | 62.98 |
| Japanese | 59.82 | 60.34 | 60.99 | 61.37 | 61.44 |
| Spanish | 58.80 | 64.32 | 68.35 | 69.34 | 69.93 |
| Mean | 63.62 | 66.83 | 69.05 | 69.84 | 70.41 |
| Czech$^\dagger$ | 76.51 | 77.48 | 78.41 | 78.59 | 78.77 |
| English$^\dagger$ | 66.04 | 67.54 | 68.37 | 69.00 | 68.96 |
| German$^\dagger$ | 41.65 | 45.94 | 46.24 | 47.45 | 47.81 |
| Mean$^\dagger$ | 61.40 | 63.65 | 64.34 | 65.01 | 65.18 |

Table 2: Semantic labelled $F_1$-scores w.r.t. training set size. $^\dagger$ indicates out-of-domain test data.

| | 10% | 25% | 50% | 75% | 100% |
|---|---|---|---|---|---|
| Catalan | 93.12 | 93.18 | 93.28 | 93.35 | 93.31 |
| Chinese | 82.37 | 82.45 | 82.54 | 82.55 | 82.57 |
| Czech | 89.03 | 89.12 | 89.17 | 89.21 | 89.20 |
| English | 87.96 | 88.38 | 88.52 | 88.67 | 88.70 |
| German | 88.23 | 89.02 | 89.63 | 89.53 | 89.64 |
| Japanese | 65.64 | 65.75 | 65.88 | 66.02 | 66.01 |
| Spanish | 93.52 | 93.49 | 93.52 | 93.53 | 93.54 |
| Mean | 85.70 | 85.91 | 86.08 | 86.12 | 86.14 |
| Czech$^\dagger$ | 86.76 | 87.02 | 87.16 | 87.08 | 87.13 |
| English$^\dagger$ | 85.67 | 86.14 | 86.22 | 86.20 | 86.23 |
| German$^\dagger$ | 77.35 | 78.31 | 79.09 | 79.10 | 79.52 |
| Mean$^\dagger$ | 83.26 | 83.82 | 84.16 | 84.13 | 84.29 |

Table 3: Semantic unlabelled $F_1$-scores w.r.t. training set size. $^\dagger$ indicates out-of-domain test data.

### 3.1 Learning rates

In order to assess the effect of training set size on semantic parsing quality, I performed a learning rate experiment, in which the proportion of the training set used for training was varied in steps between 10% and 100% of the full training set size.

Learning rates with respect to labelled $F_1$-scores are given in table 2. The improvement in scores are modest for Chinese, Czech, English and Japanese, while Catalan, German and Spanish stand out by vast improvements with additional training data. However, the improvement when going from 75% to 100% of the training data is only modest for all languages. With the exception for English, for which the parser achieves the highest score, the relative labelled $F_1$-scores follow the relative sizes of the training sets.

Looking at learning rates with respect to unlabelled $F_1$-scores, given in table 3, it is evident that adding more training data only has a minor effect on the identification of arguments.

From table 4, one can see that predicate sense disambiguation is the sub-task that benefits most from additional training data. This is not surprising, since the senses does not generalise, and hence we cannot hope to correctly label the senses of unseen predicates; the only way to improve results with the current formalism seems to be by adding more training data.

| | 10% | 25% | 50% | 75% | 100% |
|---|---|---|---|---|---|
| Catalan | 30.61 | 40.29 | 53.83 | 55.83 | 58.95 |
| Chinese | 94.06 | 94.37 | 94.71 | 95.10 | 95.26 |
| Czech | 83.24 | 84.75 | 85.78 | 86.21 | 86.60 |
| English | 92.18 | 93.68 | 94.83 | 95.35 | 95.60 |
| German | 34.91 | 47.27 | 58.18 | 62.18 | 66.55 |
| Japanese | 99.07 | 99.07 | 99.07 | 99.07 | 99.07 |
| Spanish | 38.53 | 50.22 | 59.59 | 62.01 | 66.26 |
| Mean | 67.51 | 72.81 | 78.00 | 79.39 | 81.18 |
| Czech$^\dagger$ | 89.05 | 89.88 | 91.06 | 91.38 | 91.56 |
| English$^\dagger$ | 83.64 | 84.27 | 84.83 | 85.70 | 85.94 |
| German$^\dagger$ | 33.64 | 43.36 | 42.59 | 44.44 | 45.22 |
| Mean$^\dagger$ | 68.78 | 72.51 | 72.83 | 73.84 | 74.24 |

Table 4: Predicate sense disambiguation $F_1$-scores w.r.t. training set size. $^\dagger$ indicates out-of-domain test data.

The limited power of a pipeline of local classifiers shows itself in the exact match scores, given in table 5. This problem is clearly not remedied by additional training data.

### 3.2 Dependence on syntactic parsing quality

Since I only participated in the semantic parsing task, the results reported above rely on the provided predicted syntactic dependency parsing. In order to investigate the effect of parsing quality on the current system, I performed the same learning curve experiments with gold standard parse information. These results, shown in tables 6 and 7, give an upper bound on the possible improvement of the current system by means of improved parsing quality, given that the same syntactic annotation formalism is used.

Labelled $F_1$-scores are greatly improved for all languages except for Japanese, when using gold

|  | 10% | 25% | 50% | 75% | 100% |
|---|---|---|---|---|---|
| Catalan | 6.77 | 9.08 | 11.39 | 11.17 | 12.24 |
| Chinese | 17.02 | 17.33 | 17.61 | 17.76 | 17.68 |
| Czech | 9.33 | 9.59 | 9.97 | 9.95 | 10.11 |
| English | 12.01 | 12.76 | 12.96 | 13.13 | 13.17 |
| German | 76.95 | 78.50 | 78.95 | 79.20 | 79.50 |
| Japanese | 1.20 | 1.40 | 1.80 | 1.60 | 1.60 |
| Spanish | 8.23 | 10.20 | 12.93 | 13.39 | 13.16 |
| Mean | 18.79 | 19.84 | 20.80 | 20.89 | 21.07 |
| Czech$^\dagger$ | 2.53 | 2.79 | 2.79 | 2.87 | 2.87 |
| English$^\dagger$ | 19.06 | 19.53 | 19.76 | 20.00 | 20.00 |
| German$^\dagger$ | 15.98 | 19.24 | 17.82 | 19.94 | 20.08 |
| Mean$^\dagger$ | 12.52 | 13.85 | 13.46 | 14.27 | 14.32 |

Table 5: Percentage of exactly matched predicate-argument frames w.r.t. training set size. $^\dagger$ indicates out-of-domain test data.

|  | 10% | 25% | 50% | 75% | 100% |
|---|---|---|---|---|---|
| Catalan | 62.65 | 72.50 | 75.39 | 77.03 | 78.86 |
| Chinese | 82.59 | 83.23 | 83.90 | 83.94 | 84.03 |
| Czech | 79.15 | 80.62 | 81.46 | 81.91 | 82.24 |
| English | 79.84 | 81.74 | 82.65 | 83.01 | 83.25 |
| German | 52.15 | 60.66 | 65.12 | 65.71 | 68.36 |
| Japanese | 60.85 | 61.76 | 62.55 | 62.85 | 63.23 |
| Spanish | 66.40 | 72.47 | 75.70 | 77.73 | 78.38 |
| Mean | 69.09 | 73.28 | 75.25 | 76.03 | 76.91 |
| Czech$^\dagger$ | 78.64 | 80.07 | 80.77 | 81.01 | 81.20 |
| English$^\dagger$ | 73.05 | 74.18 | 74.99 | 75.28 | 75.81 |
| German$^\dagger$ | 52.06 | 52.77 | 54.72 | 56.22 | 56.35 |
| Mean$^\dagger$ | 67.92 | 69.01 | 70.16 | 70.84 | 71.12 |

Table 6: Semantic labelled $F_1$-scores w.r.t. training set size, using gold standard syntactic and part-of-speech tag annotation. $^\dagger$ indicates out-of-domain test data.

|  | 10% | 25% | 50% | 75% | 100% |
|---|---|---|---|---|---|
| Catalan | 99.94 | 99.98 | 99.99 | 99.99 | 99.99 |
| Chinese | 92.55 | 92.67 | 92.72 | 92.63 | 92.62 |
| Czech | 91.21 | 91.27 | 91.30 | 91.30 | 91.31 |
| English | 92.34 | 92.61 | 92.85 | 92.89 | 92.95 |
| German | 93.46 | 93.59 | 94.08 | 93.85 | 94.14 |
| Japanese | 66.98 | 67.20 | 67.58 | 67.62 | 67.74 |
| Spanish | 99.99 | 99.99 | 100.00 | 100.00 | 100.00 |
| Mean | 90.92 | 91.04 | 91.22 | 91.18 | 91.25 |
| Czech$^\dagger$ | 89.00 | 89.22 | 89.34 | 89.38 | 89.36 |
| English$^\dagger$ | 92.71 | 92.56 | 92.91 | 93.06 | 93.04 |
| German$^\dagger$ | 90.54 | 90.23 | 90.77 | 90.86 | 90.99 |
| Mean$^\dagger$ | 90.75 | 90.67 | 91.01 | 91.10 | 91.13 |

Table 7: Semantic unlabelled $F_1$-scores w.r.t. training set size, using gold standard syntactic and part-of-speech tag annotation. $^\dagger$ indicates out-of-domain test data.

standard syntactic and part-of-speech annotations. For Catalan, Chinese and Spanish the improvement is in the order of 10 absolute points. For Japanese the improvement is a meagre 2 absolute points. This is not surprising given that the quality of the provided syntactic parsing was already very high for Japanese, as discussed previously.

Results with respect to unlabelled $F_1$-scores follow the same pattern as for labelled $F_1$-scores. Again, with Japanese the semantic parsing does not benefit much from better syntactic parsing quality. For Catalan and Spanish on the other hand, the identification of arguments is almost perfect with gold standard syntax. The poor labelling quality for these languages can thus not be attributed to the syntactic parse quality.

## 3.3 Computational requirements

Training and prediction times on a 2.3 GHz quad-core AMD Opteron$^{\mathrm{TM}}$system are given in table 8. Since only linear classifiers and no pair-wise feature combinations are used, training and prediction times are quite modest. Verbal and nominal predicates are trained in parallel, no additional parallelisation is employed. Most of the training time is spent on optimising the $c$ parameter of the SVM. Training times are roughly ten times as long as compared to training times with no hyper-parameter optimisation. Czech stands out as much more computationally demanding, especially in the sense disambiguation training step. The reason is the vast number of predicates in Czech compared to the other languages. The majority of the time in this step is, however, spent on writing the SVM training problems to disk.

Memory requirements range between approximately 1 Gigabytes for the smallest data sets and 6 Gigabytes for the largest data set. Memory usage could be lowered substantially by using a more compact feature dictionary. Currently every feature template / value pair is represented as a string, which is wasteful since many feature templates share the same values.

# 4 Conclusions

I have presented an effective multilingual pipelined semantic parser, using linear classifiers and a simple

| | Sense | ArgId | ArgLab | Tot | Pred |
|---|---|---|---|---|---|
| Catalan | 7m | 11m | 33m | 51m | 13s |
| Chinese | 7m | 13m | 22m | 42m | 15s |
| Czech | 10h | 1h | 1.5h | 12.5h | 34.5m |
| English | 16m | 14m | 28m | 58m | 14.5s |
| German | 4m | 2m | 5m | 13m | 3.5s |
| Japanese | 1s | 1m | 4m | 5m | 4s |
| Spanish | 10m | 16m | 40m | 1.1h | 13s |

Table 8: Training times for each language and subproblem and approximate prediction times. Columns: training times for sense disambiguation (Sense), argument identification (ArgId), argument labelling (ArgLab), total training time (Tot), and total prediction time (Pred). Training times are measured w.r.t. to the union of the official training and development data sets. Prediction times are measured w.r.t. to the official evaluation data sets.

greedy constraint satisfaction heuristic. While the semantic parsing results in these experiments fail to reach the best results given by other experiments, the parser quickly delivers quite accurate semantic parsing of Catalan, Chinese, Czech, English, German, Japanese and Spanish.

Optimising the hyper-parameters of each of the individual classifiers is essential for obtaining good results with this simple architecture. Syntactic parsing quality has a large impact on the quality of the semantic parsing; a problem that is not remedied by adding additional training data.

## References

Aljoscha Burchardt, 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 International Conference on Language Resources and Evaluation (LREC-2006)*, Genoa, Italy.

Koby Crammer and Yoram Singer. 2002. On the learnability and design of output codes for multiclass problems. *Machine Learning*, 47(2):201–233, May.

Rong-En Fan, 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.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288.

Jan Hajič, Jarmila Panevová, Eva Hajičová, Petr Sgall, Petr Pajas, Jan Štěpánek, Jiří Havelka, Marie Mikulová, and Zdeněk Žabokrtský. 2006. Prague Dependency Treebank 2.0. CD-ROM, Cat. No. LDC2006T01, ISBN 1-58563-370-4, Linguistic Data Consortium, Philadelphia, Pennsylvania, USA.

Jan Hajič, 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 the 13th Conference on Computational Natural Language Learning (CoNLL-2009), June 4-5*, Boulder, Colorado, USA.

Richard Johansson and Pierre Nugues. 2008. Dependency-based syntactic–semantic analysis with PropBank and NomBank. In *Proceedings of the Shared Task Session of CoNLL-2008*, Manchester, UK.

Daisuke Kawahara, Sadao Kurohashi, and Kôiti Hasida. 2002. Construction of a Japanese relevance-tagged corpus. In *Proceedings of the 3rd International Conference on Language Resources and Evaluation (LREC-2002)*, pages 2008–2013, Las Palmas, Canary Islands.

Andrew Kachites McCallum. 2002. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu.

Martha Palmer and Nianwen Xue. 2009. Adding semantic roles to the Chinese Treebank. *Natural Language Engineering*, 15(1):143–172.

Yvonne Samuelsson, Oscar Täckström, Sumithra Velupillai, Johan Eklund, Mark Fishel, and Markus Saers. 2008. Mixing and blending syntactic and semantic dependencies. In *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*, pages 248–252, Manchester, England, August. Coling 2008 Organizing Committee.

Mihai Surdeanu, 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 the 12th Conference on Computational Natural Language Learning (CoNLL-2008)*, Manchester, Great Britain.

Mariona Taulé, Maria Antònia Martí, and Marta Recasens. 2008. AnCora: Multilevel Annotated Corpora for Catalan and Spanish. In *Proceedings of the 6th International Conference on Language Resources and Evaluation (LREC-2008)*, Marrakesh, Morroco.

Nianwen Xue and Martha Palmer. 2004. Calibrating features for semantic role labeling. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 88–94, Barcelona, Spain, July. Association for Computational Linguistics.