# UBC-ALM: Combining k-NN with SVD for WSD

**Eneko Agirre and Oier Lopez de Lacalle**
IXA NLP Group
University of the Basque Country
Donostia, Basque Country
{e.agirre,jibloleo}@ehu.es

## Abstract

This work describes the University of the Basque Country system (UBC-ALM) for lexical sample and all-words WSD subtasks of SemEval-2007 task 17, where it performed in the second and fifth positions respectively. The system is based on a combination of $k$-Nearest Neighbor classifiers, with each classifier learning from a distinct set of features: local features (syntactic, collocations features), topical features (bag-of-words, domain information) and latent features learned from a reduced space using Singular Value Decomposition.

## 1 Introduction

Our group (UBC-ALM) participated in the lexical sample and all-words WSD subtasks of SemEval-2007 task 17. We applied a combination of different $k$-Nearest Neighbor ($k$-NN) classifiers. Each classifier manages different information sources (features), making the combination a powerful solution. This algorithm was previously tested on the datasets from previous editions of Senseval (Agirre et al., 2005; Agirre et al., 2006). Before submission, the performance of the system was tested on the SemEval lexical sample training data. For learning we use a rich set of features, including latent features obtained from a reduced space using Singular Value Decomposition (SVD).

This paper is organized as follows. The learning features are presented in section 2, and the learning algorithm and the combinations of single $k$-NNs are given in section 3. Section 4 focuses on the tuning experiments. Finally, section 5 summarizes the official results and some conclusions.

## 2 Feature set

We relied on an extensive set of features of different types, obtained by means of different tools and resources. We defined two main groups: the **original features** extracted directly from the text, and the **SVD features** obtained after applying SVD decomposition and projecting the original features into the new semantic space (Agirre et al., 2005).

### 2.1 Original features

**Local collocations**: bigrams and trigrams formed with the words around the target. These features are constituted by lemmas, word-forms, or PoS tags[1]. Other local features are those formed with the previous/posterior lemma/word-form in the context.

**Syntactic dependencies**: syntactic dependencies were extracted using heuristic patterns, and regular expressions defined with the PoS tags around the target[2]. The following relations were used: object, subject, noun-modifier, preposition, and sibling.

**Bag-of-words features**: we extract the lemmas of the content words in the whole context, and in a ±4-word window around the target. We also obtain salient bigrams in the context, with the methods and the software described in (Pedersen, 2001).

**Domain features**: The WordNet Domains resource was used to identify the most relevant domains in the context. Following the relevance formula presented in (Magnini and Cavagliá, 2000), we defined 2 feature types: (1) the most relevant domain, and (2) a list of domains above a predefined threshold[3].

[1] The PoS tagging was performed with the fnTBL toolkit (Ngai and Florian, 2001).

[2] This software was kindly provided by David Yarowsky's group, from Johns Hopkins University.

[3] The software to obtain the relevant domains was kindly provided by Gerard Escudero's group, from Universitat Politec-

### 2.2 SVD features

Singular Value Decomposition (SVD) is an interesting solution to the sparse data problem. This technique reduces the dimensions of the vectorial space finding correlations and collapsing features. It also gives the chance to use unlabeled data as an additional source of correlations.

$M \ni \mathbf{R}^{m \times n}$, a matrix of features-by-document is built from the training corpus and decomposed into three matrices, as shown in Eq. (1). $U$ and $V$, row and column matrix, respectively, have orthonormal columns and $\Sigma$ is a diagonal matrix which contains $k$ eigenvalues in descending order.

$$M = U \Sigma V^T = \sum_{i=1}^{k=min\{m,n\}} \sigma_i u_i vi^T \qquad (1)$$

We used the *singular value* matrix ($\Sigma$) and the *column* matrix ($U$) to create a projection matrix, which is used to project the data (represented in features vectors) from the original space to a reduced space. Prior to that we selected the first $p$ columns from the $\Sigma$ and $U$ matrices ($p < k$): $\vec{t_p} = \vec{t}^T U_p \Sigma_p^{-1}$

We have explored two different variants in order to build a matrix, and obtain the SVD features:

**SVD One Matrix per Target word (SVD-OMT)**. For each word (i) we extracted all the features from the given training (test) corpus, (ii) built the feature-by-document matrix from training corpus, (iii) decomposed it with SVD, and (iv) project all the training (test) data. Note that this variant has been only used in the lexical sample task due to its costly computational requirements.

**SVD Single Matrix for All target words (SVD-SMA)**: (i) we extracted bag-of-words features from the British National Corpus (BNC) (Leech, 1992), (ii) built the feature-by-document matrix, (iii) decompose it with SVD, and (iv) project all the data (train/test).

## 3 Learning Algorithm

The machine learning (ML) algorithm presented in this section rely on the previously described features. Each occurrence or instance is represented by the features found in the context $(f_i)$. Given an occurrence of a word, the ML method below returns a weight for each sense $(weight(s_k))$. The sense with maximum weight will be selected.

We use a set of combination of the $k$-**Nearest Neighbor** ($k$-NN) to tag the target words in both the lexical sample and all-words tasks.

### 3.1 $k$-Nearest Neighbor

$k$-NN is a memory-based learning method, where the neighbors are the $k$ most similar contexts, represented by feature vectors ($\vec{c_i}$), of the test vector ($\vec{f}$). The similarity among instances is measured by the cosine of their vectors. The test instance is labeled with the sense obtaining the maximum sum of the weighted votes of the $k$ most similar contexts. The vote is weighted depending on its (neighbor) position in the ordered rank, with the closest being first. Eq. (2) formalizes $k$-NN, where $C_i$ corresponds to the sense label of the $i$-th closest neighbor.

$$\arg\max_{S_j} = \sum_{i=1}^{k} \begin{cases} \frac{1}{i} & \text{if } C_i = S_j \\ 0 & \text{otherwise} \end{cases} \qquad (2)$$

### 3.2 $k$-NN combinations and feature splits

As seen in section 2 we use a variety of heterogeneous sets of features. Our previous experience has shown that splitting the problem up into more coherent spaces, training different classifiers in each feature space, and then combining them into a single classifier is a good way to improve the results (Agirre et al., 2005; Agirre et al., 2006). Depending on the feature type (original features or features extracted from SVD projection) we split different sets of feature spaces. In total we tried 10 features spaces.

For the **original features**:

- **all_feats**: Extracted all original features.
- **all_notdom**: All original features except domain features.
- **local**: All the original features except domain and bag-of-words features.
- **topic**: The sum of bag-of-words and domain features.
- **bow**: Bag-of-word features.
- **dom**: Domain features.

nica de Catalunya

| **Combination** | **accuracy** |
|---|---|
| all_feats+topic+local+SVD-OMT[all_feats]+SVD-OMT[topic]+SVD-OMT[local] | 88.8 |
| all_feats+all_notdom+topic+local+SVD-SMA+SVD-OMT[all_feats]+SVD-OMT[topic]+SVD-OMT[local] | 88.7 |
| all_feats+topic+local+SVD-SMA+SVD-OMT[all_feats]+SVD-OMT[topic]+SVD-OMT[local] | 88.5 |
| all_notdom+topic+local+SVD-SMA+SVD-OMT[all_feats]+SVD-OMT[topic]+SVD-OMT[local] | 88.5 |
| all_feats+all_notdom+topic+local | 88.4 |
| all_notdom+local+SVD-SMA | 88.3 |
| all_feats+all_notdom+local+SVD-SMA | 88.2 |
| all_notdom+topic+local | 88.1 |
| all_feats+topic+local | 88.1 |
| **word-by-word optimization** | **89.5** |

Table 1: Result for the best $k$-NN combinations in 3 fold cross-validation SemEval lexical sample.

For the **SVD features**:

- **SVD-OMT[all_feats]**: OMT matrix applied to all original features.
- **SVD-OMT[local]**: OMT matrix to the **local** original features.
- **SVD-OMT[topic]**: OMT matrix to the **topic** original features.
- **SVD-SMA**: Features obtained from the projection of **bow** features with the SMA matrix.

Depending on the ML method one can try different approaches to combine classifiers. In this work, we exploited the fact that a $k$-NN classifier can be seen as $k$ points casting each one vote. The votes are weigthed by the inverse ratio of its position in the rank $(k - r_i + 1)/k$, where $r_i$ is the rank. Each of the $k$-NN classifiers is trained on a different feature space and then combined.

## 4 Experiments on training data

We optimized and tuned the system differently for each kind of tasks. We will examine each in turn.

### 4.1 Optimization for the lexical sample task

For the lexical sample task we only use the training data provided. We tuned the classifiers using 3 fold cross-validation on the SemEval lexical sample training data. We tried to optimize several parameters: number of neighbors, SVD dimensions and best combination of the single $k$-NNs. We set $k$ as one of $1, 3, 5$ and 7, and the SVD dimension ($d$) as one of $50, 100, 200$ and 300. We also fixed the best combination. This is the optimization procedure we followed:

1. For each single classifier and feature set (see section 2), check each parameter combination.
2. Fix the parameters for each single classifier. In our case, $k = 5$ and $k = 7$ had similar results, so we postponed the decision. $d = 200$ was the best dimension for all classifiers, except SVD-OMT[topic] which was $d = 50$.
3. For the best parameter settings ($k = 5; k = 7$ and $d = 200$; $d = 50$ when SVD-OMT[topic]) make *a priori* meaningful combinations (due to CPU requirements, not all combination were feasible).
4. Choose the $x$ best combination overall, and optimize word by word among these combination. We set $x = 8$ for this work, $k$ was fixed in $5$, and $d = 200$ (except with SVD-OMT[topic] which was $d = 50$).

Table 1 shows the best results for 3 fold cross-validation in SemEval lexical sample training corpus. The figures show that optimizing each word the performance increases 0.7 percentage points over the best combination.

### 4.2 Optimization for the all-words task

To train the classifiers for the all-words task we just used Semcor (Miller et al., 1993). In (Agirre et al., 2006) we already tested our approach on the Senseval-3 all-words task. The best performance for the Senseval-3 all-words task was obtained with $k = 5$ and $d = 200$, but we decided to to perform further experiments to search for the best combination. We tested the performance of the combination of single $k$-NN training on Semcor and testing both on the Senseval-3 all-words data (cf. Table 2) and on the training data from SemEval-2007 lexical sample (cf. Table 3).

Note that tables 2 and 3 show contradictory results. Given that in SemEval-2007 lexical sample

| Combination | rec. | prec. |
|---|---|---|
| all_feats+local+notbow | 0.685 | 0.685 |
| all_feats+local+SVD-SMA | 0.679 | 0.679 |
| all_feats+topic+local+SVD-SMA | 0.689 | 0.689 |

Table 2: Results for the best $k$-NN combinations in Senseval-3 all-words, using Semcor as training corpus.

| Combination | rec. | prec. |
|---|---|---|
| all_feats+SVD-SMA | 0.666 | 0.666 |
| all_feats+local+SVD-SMA | 0.661 | 0.661 |
| all_feats+topic+local+SVD-SMA | 0.664 | 0.664 |

Table 3: Results for the best $k$-NN combinations in training part of SemEval lexical sample, using Semcor as training corpus.

| Task | Method | Rank | rec. | prec. |
|---|---|---|---|---|
| LS | Best | 1 | 0.887 | 0.887 |
| LS | UBC-ALM | 2 | 0.869 | 0.869 |
| LS | Baseline | - | 0.780 | 0.780 |
| AW | Best | 1 | 0.591 | 0.591 |
| AW | k-NN combination | 5 | 0.544 | 0.544 |
| AW | Baseline | - | 0.514 | 0.514 |

Table 4: Official results for SemEval-2007 task 17 lexical sample and all-words subtasks.

the senses are more coarse grained, we decided to take the best combination on Senseval-3 all-words for the final submission.

## 5 Results and conclusions

Table 4 shows the performance obtained by our system and the winning systems in the SemEval lexical sample and all-words evaluation. On the lexical sample evaluation our system is $2.6$ lower than the cross-validation evaluation. This can be a sign of a slight overfitting on the training data. All in all we ranked second over 13 systems.

Our all-words system did not perform so well. Our system is around $4.7$ points below the winning system, ranking 5th from a total of 14, and 3 points above the baseline given by the organizers. This is a disappointing result when compared to our previous work on Senseval-3 all-words where we were able to beat the best official results (Agirre et al., 2006). Note that the test set was rather small, with 465 occurrences only, which might indicate that the performance differences are not statistically significant. We plan to further investigate the reasons for our results.

## Acknowledgments

We wish to thank to David Martínez for helping us extracting learning features. This work has been partially funded by the Spanish education ministry (project KNOW). Oier Lopez de Lacalle is supported by a PhD grant from the Basque Government.

## References

E. Agirre, O.Lopez de Lacalle, and David Martínez. 2005. Exploring feature spaces with svd and unlabeled data for Word Sense Disambiguation. In *Proceedings of the Conference on Recent Advances on Natural Language Processing (RANLP'05)*, Borovets, Bulgaria.

E. Agirre, O. Lopez de Lacalle, and D. Martínez. 2006. Exploring feature spaces with svd and unlabeled data for Word Sense Disambiguation. In *Proceedings of the XXII Conference of Sociedad Espaola para el Procesamiento del Lenguaje Natural (SEPLN'06)*, Zaragoza, Spain.

G. Leech. 1992. 100 million words of English: the British National Corpus. *Language Research*, 28(1):1–13.

B. Magnini and G. Cavagliá. 2000. Integrating subject field codes into WordNet. In *Proceedings of the Second International LREC Conference*, Athens, Greece.

G.A. Miller, C. Leacock, R. Tengi, and R.Bunker. 1993. A Semantic Concordance. In *Proceedings of the ARPA Human Language Technology Workshop. Distributed as* Human Language Technology *by San Mateo, CA: Morgan Kaufmann Publishers.*, pages 303–308, Princeton, NJ.

G. Ngai and R. Florian. 2001. Transformation-Based Learning in the Fast Lane. *Proceedings of the Second Conference of the North American Chapter of the Association for Computational Linguistics, pages 40-47, Pittsburgh, PA, USA.*

T. Pedersen. 2001. A Decision Tree of Bigrams is an Accurate Predictor of Word Sense. In *Proceedings of the Second Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL-01)*, Pittsburgh, PA.