Defining Classifier Regions for WSD Ensembles using Word Space Features

Harri M.T. Saarikoski[1], Steve Legrand[2,3], and Alexander Gelbukh[3]

[1] KIT Language Technology Doctorate School, Helsinki University, Finland
Harri.Saarikoski@helsinki.fi

[2] Department of Computer Science, University of Jyväskylä, Finland
stelegra@cc.jyu.fi

[3] Instituto Politecnico Nacional, Mexico City, Mexico
gelbukh@gelbukh.com

Abstract. Based on recent evaluation of word sense disambiguation (WSD) systems [10], disambiguation methods have reached a standstill. In [10] we showed that it is possible to predict the best system for target word using word features and that using this 'optimal ensembling method' more accurate WSD ensembles can be built (3-5% over Senseval state of the art systems with the same amount of possible potential remaining). In the interest of developing if more accurate ensembles, w e here define the strong regions for three popular and effective classifiers used for WSD task (Naive Bayes - NB, Support Vector Machine - SVM, Decision Rules - D) using word features (word grain, amount of positive and negative training examples, dominant sense ratio). We also discuss the effect of remaining factors (feature-based).

1 Introduction

Numerous methods of disambiguation have been tried to solve the WSD task [1,8] but no single system or system type (e.g. classifier) has been found to perform superiorly for all target words. The first conclusion from this is that different disambiguation methods result in different performance results. System bias is the inherent and unique capability or tendency of the classifier algorithm to transform training data into a useful sense decision model. A second conclusion is that there is a 'word bias', i.e. each word poses a different set of learning problems. Word bias is the combination of factors particular to that word that cause classification systems to vary their performance considerably. Differences of up to 30% in precision at word can occur even with top systems.

Optimal ensembling method is dedicated to mutually solve these two biases, to map a particular type of system to a particular type of word. It attempts first of all to discover *n* base systems which are as strong and complementary (with regard to performance) as possible and then train itself using training words to recognize which system will be strongest for a given test word. Optimal ensembles have largely been neglected in WSD in favor of single-classifier systems, trained on the same feature set (e.g. [7,12]) or 'conservative ensembles' (e.g. voting pool of six base systems [9]) where the same system(s) is applied for all test words. It is reasonable to assume that system (and particularly its classifier algorithm) strengths tend to follow changes (drops and rises) in the details of each learning task (i.e. ambiguous word). This assumption was proven correct by [13] who showed that systems differ in different regions of word grain, amount of training and most frequent (dominant) sense bias. According to [13], one base system typically excels in the lower and higher regions of a factor and another in the middle region (e.g. NB systems in grain region 12..22 while a transformation-based learner, TBL, thrived in the surrounding regions ..12 and 22.. [13]). Effect of classifier selection on classification system performance has been reported in numerous works [2,9,12,14]. For instance, [2] studied the effect of skewing the training distribution on classifier performance. They found three classifiers (Naive Bayes or NB, SVM, Multinomial Naive Bayes) to occupy different but intact regions in word space.

In [10] we presented the method of optimal ensembling of any base systems. The method specifies how we can discover the base systems whose strengths at different learning tasks (words in WSD) complement each other. In this paper, we attempt to further generalize the effect of classifier selection on the strong region of the system using various combinations of three word factors (grain, positive vs negative examples per sense, dominant vs sub-dominant sense ratio). We present two sets of experiments using Senseval-2 and Senseval-3 English lexical sample datasets.

In Section 2, we present the machine-learning tools we used for performing the system analyses and prediction tests. In section 3, we discuss the word and system factors we used for predictions. In section 4, we visualize some of the training models to be used by predictors. Final sections 5 and 6 are dedicated to discussions and conclusions.

2 Tools

Study of disambiguation systems lacks a diagnostic tool that could be used to meta-learn the effects of these factors. As a result, the following types of questions are largely unanswered: Which are the words where a system is at its strongest? What type of ensembles of systems achieve optimum performance for give target word?

We are developing a meta-classifier (MOA-SOM, 'mother-of-all-self-organizing-maps') to handle such learning tasks. The tool clusters publicly available WSD system scores [1,8] stored in database [10] based on features defining the systems (e.g. classifier algorithm, feature sets) and target words (e.g. PoS, training, word grain) by calculating the amount of correlation between systems and words. The output from MOA-SOM is the optimal classifier, feature and configuration for that target word. The feature matrix can be fed to SOM using either system names as labels and words as data points or vice versa. The SOM used is based on hierarchically clustering DGSOT [5] which was found useful in earlier WSD experiments [4]. For these tests we additionally employed the machine-learning algorithms implemented in Weka toolkit [11] for training and testing the predictors and YALE toolkit [6] for visualizing the system regions / training models.

In the next section, we define the method of optimal ensembling that was first introduced in [10].

3 Method

3.1 Motivation for Factors -> Predictors

The motivation behind optimal ensembles is that classifiers have inherently different solutions to deal with the different learning tasks (and processing the training data). In [10] we showed how word factors (e.g. grain) can be used to build a predictor of best system for given target word. Interestingly, it seems that a good system *difference* predictor will also need to be an excellent system *performance* predictor, i.e. predictor of the accuracy of the best base system in the ensemble. Our best predictors in [10] that obtained a high of 0.85 prediction accuracy were correlated strongly or very strongly with base system performance (using Spearman's correlation coefficient we got a high of 0.88 correlation). Furthermore, we found that the very easy or very hard words exhibited little if at all difference between systems.

Based on this intimate correlation, we can draw a schematic (Figure 1) that represents the regions in word space where the biggest vs smallest differences between systems take place (see Figures 2 and 3).

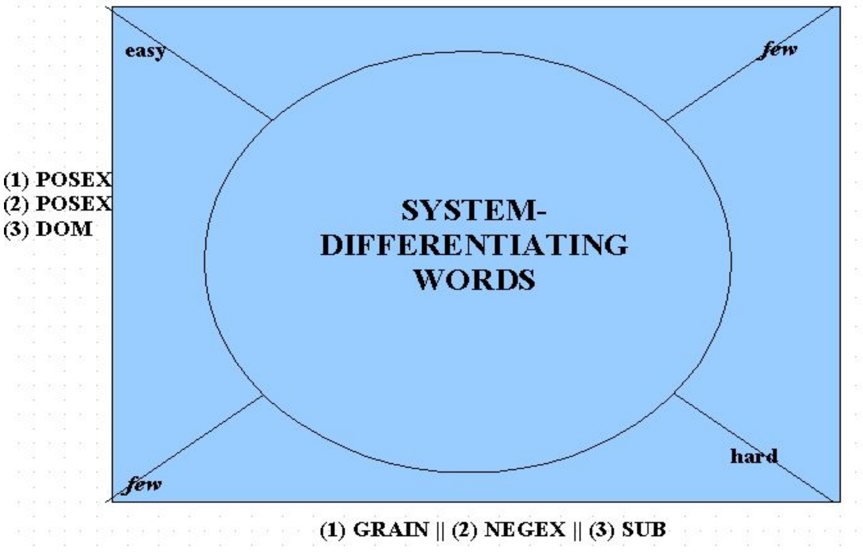

Figure 1. System-differentiation potential in three factor pairs representing word space - (1) posex-grain, (2) posex-negex and (3) dom-sub. *Posex* is the average number of training instances per sense and *negex* the *average* number of training examples per sense-class. *Dom* and *sub* represent the training distribution between dominant (most frequent) sense and subdominant (next most frequent) sense. *Grain* is obviously the number of senses in the sense repository used in Senseval evaluations (usually WordNet).

In Figure 1, top left corner (e.g. low-grain, high-trainword) features the *easy* words that basically any system can disambiguate to highest accuracy (e.g. *graceful[a]* in Senseval-2). Bottom right corner (e.g. low-posex, high-negex) contains the *hard* words (e.g. *draw[v]* in Senseval-2) that systems find equally hard to disambiguate (typically disambiguation accuracy remains below 50%). The corners marked *few* contain very few words falling into those regions, at least in Senseval evaluations.

In our experiments [10], we largely ignored words in *easy, hard* and *few* regions from our training and testing data as well as and systems whose strength is focused in those regions. Instead we focused on the center region (*System-differentiating words*) where systems exhibit greatest differences in performance. This is simply because the feasibility of net gain by an optimal ensemble over base systems is at its greatest in that center region.

3.2 Factors

We introduce here the three word-based factors in explaining variations in system performance (Train, Grain, and DomSub). *Train* is average number of training instances per sense, *Grain* is the number of senses (as recorded in WordNet / WordSmyth sense repositories used in Senseval evaluations).

3.3 Predictors

A few factor formulas emerged as best predictors of system difference predictors. To train the predictors, we used both manual rules and machine-learning algorithms:

(1) Bisections (baseline). To achieve a bisection baseline, we first sort the data according to a selected factor (e.g. T, G, D, T+G+D), then split the data in two and calculate the net gain by each system for each half and average that by dividing it by two. The best weighting scheme we found was the square root of the unweighted sum of normalized values of the three factors: `sqrt (a*T + b*G + c*D)` where *G* stands for Grain, *T* for Train, *D* for DomSub values of target words and integers *a*, *b* and *c* normalize the weights of the three factors. Note that since this set of predictors is limited to one factor at a time, it cannot express decision rules containing multiple factors which tends to make them less reliable.

(2) Machine-learned models. T o predict the best system for words, we trained some of the most efficient learning algorithms implemented in Weka toolkit [16] (Support Vector Machine, Maximum Entropy, Naive Bayes, Decision Trees, Random Forests as well as voting committee, training data bagging and algorithm boosting methods). For training we used the abovementioned word factors both individually and in various permutations (e.g. T-G).

3.4 Optimal Ensembling Method Embedded in a WSD Algorithm

In this section, we outline a method for defining and selecting maximally complementary base systems integrated inside a disambiguation algorithm:

- **Base system selection.** Run candidate base systems on training words. Investigate their performance at different types of words. Based on their performance at training words, select systems whose strong regions are as large and as distinct as possible, i.e. maximally complementary, using the following criteria:
  - biggest gross gain (defined in Evaluation below) of the constructed optimal ensemble over better of candidate base systems
  - largest number of training words won by the system
  - largest strong region define in word spaces define
  - two base systems with a large complementary nature
- **Training the predictor.** Using the training run data, train the predictors to recognize the best base system using readily available factors (e.g. word grain). Predictor can be constructed by setting decision rules manually, e.g. "use system#1 (Decision Tree -based) when number of senses (grain) < 5, system#2 (Naive Bayes -based) when grain is > 5 but not when 20 < train < 25". Alternatively, use a machine-learning algorithm to induce the rules from the training data.
  - In order to *see* maximal complementarity of selected base systems in word space, use drawing of strong regions of base systems, a visualization of the predictor training model (see Figures)
- **Testing.** Run selected base systems and the optimal ensemble according to the system-selection rules set by the best predictor on test words.
- **Evaluation.** Evaluate the performance of the optimal ensemble by comparing it to the better of the base systems. Also evaluate the predictor using *net gain* measure calculated from the following formula:

  ```
  ((PredictionAccuracy - (1.0 / NumberOfSystems)) *2)
                     * GrossGain
  ```

  PredictionAccuracy is the number of correct system-for-word predictions out of all test words and *NumberOfSystems* is the number of classes/systems to predict. *GrossGain* is a measure of the potential of the base systems when they form an ensemble, resulting from a perfect system-for-word prediction for all test words. It is calculated from all-words average net gain by either base system (e.g. in a test set of two words, if system#1 wins over system#2 by 2% at word#1 and system#2 wins over system#1 by 4% at word#2, then gross gain for all test words is (2+4) / 2 = 3%). *Net gain* is then calculated as follows: in a two-system ensemble with 0.80 prediction accuracy and 8.0% gross gain, net gain is ((0.80-0.50)*2)) * 8.0% = 4.8%.
- **Development**: Predictors and base systems should be developed together. Therefore, development of optimal ensembles can start either from good predictors or from good base systems:
  - Keep the ensemble with biggest net gain and try to find a better predictor of best system, altering learning algorithm and/or word factors

(e.g. weighting)

- Keep the ensemble with the best predictor and alter the base systems (make one or several of them stronger) so that a bigger net gain results.

4 Tests

In this section we describe the prediction experiments of SVM/NB based systems in two WSD datasets (Senseval-2 and Senseval-3 English lexical sample).

4.1 Test Setting

We investigated the following factor pairs or word spaces (posex-negex, posex-grain and dom-sub) to define the strong regions of systems based on three classifiers (SVM/NB/D[1]) **in Senseval-2 and Senseval-3 English lexical sample evaluation datasets:**

Table 1. Systems based on the three classifiers in two datasets.

Dataset	Classifier	System names
Senseval-2	SVM	UMCP
	NB	Duluth1, Duluth4
	D	TALP (boosted), DuluthB, Duluth5, Duluth2, Duluth3 (multi)
Senseval-3	SVM	IRST-kernel, nusels, TALP, UMCP
	NB	htsa3, CLaC1, Prob1
	D	SyntaLex3 (multi), Duluth-ELSS (multi)[2]

Two thirds of available word set was used for training the predictor model, and the remaining one third was used for testing the model. In the following box plot (Figure 2) we see the word factor values for those base systems we are investigating.

[1] *D* stands for decision rule based classifiers (decision trees, decision lists, decision stumps).

[2] *Multi* signifies that several decision tree classifiers using different feature sets were bagged and a committee decision rendered. *Boosted* signifies that the classifier employed boosting technique (AdaBoost).

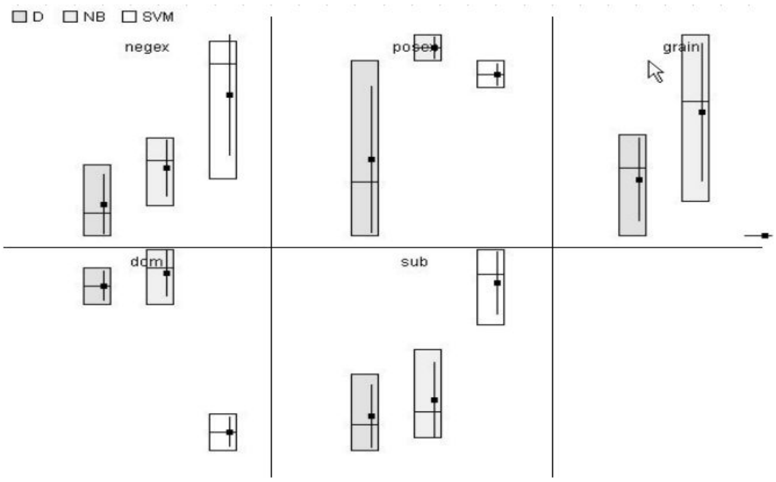

Figure 2. Box plots showing five word factors (*negex, posex, grain, dom, sub*) for D, NB and SVM systems in Senseval-3. Box plot features the following information: length of the column is the amount of *variation* of values, and the vertical line running through that column indicates actual *maximum* and *minimum* values in the dataset. Square dot in the middle of the column is the *average* of values, horizontal line in the vicinity of that is the *median* member of the dataset.

In Figure 2, we can see SVM, NB and D based systems differing in practically all factors. Specifically, the system region cores (dot inside the column) are very different and also the variation (range of the filled column) indicating the borders of its strong region.

Let us now look at the system-differentiating capability of a few factors in detail.

4.2 Strong Regions of Classifiers

Positive vs negative examples per sense. [2] used negex-posex space to illuminate the fundamental difference of SVM vs NB classifiers in a text categorization task. Let us see whether that space is equally effective discriminator for WSD systems.

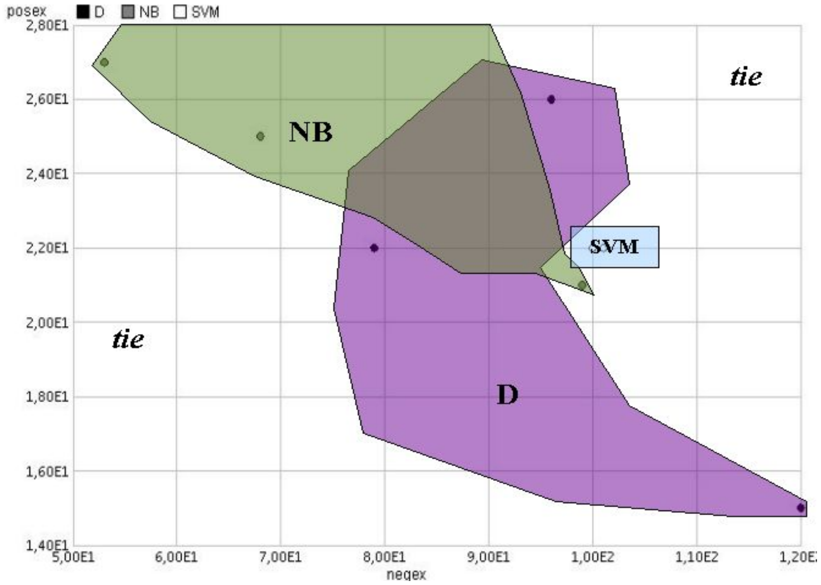

Figure 3. SVM/NB/D regions (posex-negex space) in Senseval-2.

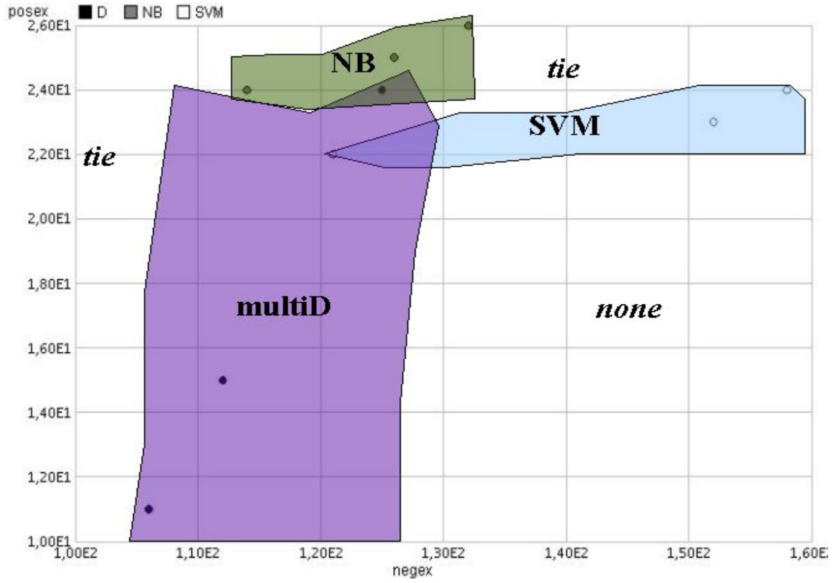

Figure 4. SVM/NB/D regions (posex-negex space) in Senseval-3. To draw the system regions in the following figures, we calculated the *average* values for word factors from bins of 10, 20 and 30 best words by each system. i.e. this is box plot data in two dimensions with selected two factors. Scale values in the figures (produced with YALE toolkit [6]) are read as follows: e.g. *posex* **value of 2*.20E2* means average of 220 positive examples per word sense.**

Looking very closely at Figures 3 and 4, we can see approximate resemblance (in shape, size and orientation) of the strong regions of three classifiers in two different datasets. In particular, we want to point out the placement of SVM, NB and D regions relative to each other (NB strongest in high-posex, D in low-posex and SVM strongest in high-negex, D systems were strong in high-negex, low-posex region.) Also notice the overlapping of the regions. Those are the regions where systems are most likely to be tied, i.e. have equal performance.

Grain and dominant sense. In order to define the strong regions more accurately, we need to look at other factors as well (especially word grain and dominant sense ratio). In [10] we showed that train and grain factors quite well differentiate JHU and SMU, i.e. that SMU [7] was a low-grain (high-posex) and JHU [12] a high-grain (low-posex) word expert. For example, with verbs such as *call* and *carry* with 30-40 senses and less than 10 training examples per sense, the winning margin of JHU over SMU is at its greatest, while with nouns such as *sense* and *art* with 4-5 senses and more than 25 training examples per sense, SMU was better.

In the current experiment with Senseval-3 systems, we found that NB strength is focused on high-grain region (core at grain=11) while the whole SVM region is set around grain=7. As to dominant sense ratio, we found SVM/NB (and D) regions in Senseval-3 to follow largely the same borderlines as posex-negex map (Figure 1). This was rather expected due to the significant correlation between *average* number of training examples per sense (posex) vs per *dominant* sense (dom). In sum, NB excelled at high-dom, low-sub while SVM at low(er)-dom and high(er)-sub.

5 Results

Table 2. Results from applying the method on selected base systems from Senseval-3

base systems (gross gain)	prediction accuracy[3]	net gain (ensemble over base systems)
htsa3+IRST-kernel (4.1%)	**0.82**	**2.7%**
htsa3+nusels (3.6%)	0.70	1.4%
nusels+ IRST-kernel (**4.4%**)	0.80	2.6%
htsa3+IRSTk+nusels (6.1%)	0.55	**2.7%**
SVMall + NBall (3.8%)	0.73	1.7%

Table 2 shows nusels+IK is the maximally complementary system pair in terms of net gain but another system pair (nusels+IRST-kernel) has the higher potential (gross gain). It should also be noted that the more challenging three-system prediction task

[3] *Prediction accuracy* of 0.85, for example, means that the best (better) base system was predicted right for 85 out of 100 test words.

(htsa3+IRSTk+nusels) produces equally high net gain as htsa3+IRST-kernel pair.

Table 3. Results from applying the method on selected base systems from Senseval-2

base systems (gross gain)	prediction accuracy	net gain (ensemble over base systems)
JHU+SMU (8.0%)	0.80	4.8**%**
SMU+KUN (**8.4%**)	**0.85**	**5.1%**
JHU+KUN (5.5%)	0.75	2.8%
JHU+SMU+KUN (9.5%)	0.55	4.2%
SVMall+NBall (+++)	s2 easy but few	

According to Table 3, SMU+KUN appears to have the highest gross gain, prediction accuracy and net gain, making it the maximally complementary system pair for this dataset. Furthermore, it seems that the more difficult 3-system prediction (JHU+SMU+KUN) with more gross gain loses to 2-system predictions in prediction accuracy ending up with a slightly lower net gain.

Predictions. Best predictive factors (and learners) in all experiments turned out to vary according to base system pair. The most reliable learning algorithms for best-system prediction turned out to be Support Vector Machines and slightly less consistently Maximum Entropy and Naive Bayes classifiers. Machine-learning models (2) tend to work better than the corresponding bisection baseline (1). We eliminated each factor in turn from the training model to look at the contribution of the factors. The contribution of individual factors to system differentiation seems to depend heavily on the base system pair. Prediction power of the individual factors varied between 0.60 and 0.80. A combination of factors to define the strong region tended to work better than individual factors (e.g. posex+grain+domsub for SMU/JHU pair).

(*) Somewhat (un)expectedly predicting between SVM and NB clusters proved to be harder than between individual systems. A cluster is an compound averaged from several individuals who (while sharing one factor) exhibit considerable differences. This prediction of SVM/NB was not meant for optimal ensembling, rather to define core region of those classifiers. it makes no sense to predict between clusters until clusters are adequately defined (missing feature-level factors).

6 Discussion

Systems based on various classifiers (SVM, NB and D) appear to occupy quite different regions in word space as they did for text categorization systems in [2]. The respective placements of SVM and NB in our data (Figures 2 and 3) are not the same in [2] due to different task settings but some similarity can be found: SVM is set in the middle posex region with a higher negex, NB immediately over it at lower negex. D systems occupy the high-posex region.

Supporting evidence of the inherent difference of classifier on strong region can be found. First, Duluth systems in Senseval-2 [9]. We compared these 'minimal pair' systems (NB or D based) in various word spaces (negex-posex, posex-grain, dom-sub). Duluth best 5 is 14% off best, rest 16%. Second, when looking at instance-based classifiers (SMU, GAMBL) in Senseval-2 and Senseval-3 evaluations (respectively). In both evaluations, this simple classifier is strongest at low-grain, high-train region of word space. It seems evident that systems with fairly 'simple' classifiers (Decision Stump in [9], Transformation-Based Learner in [13], SMU system in [10]) perform well with words in the *easy* region (top left corner in Figure 1). On the other hand, more complex classifiers (e.g. NB and SVM and multi-system ensembles) are more resilient to e.g. lack of training associated with high-grain words, and therefore find their core strength in the opposite corner (bottom right) of word spaces in Figure 1.

7 Conclusions and Future

We have elaborated on a method for defining the strong regions of WSD systems using a combination of known and readily available word factors. We can conclude that selection of classifier sets the approximate core of a WSD system's strong region. We found the relative strength of two most popular classifiers in WSD (Naive Bayes and Support Vector Machine) to complement each other in terms of almost all the word spaces investigated. It can now also be better understood why these two classifiers are the most popular ones experimented for WSD task: they command large but non-overlapping regions over other classifiers, i.e. disambiguate large numbers of target words to high(est) accuracy.

With a fully correct prediction of best system for all words (best prediction currently is 0.85), the method has the potential to raise state-of-the-art accuracy of WSD systems considerably more than a few percentages. We consider the remaining misclassifications (15 out of 100 test words) to be primarily due to inadequate accounting of feature-level factors: *number* of feature sets (e.g. 1-grams as opposed to 1-grams and 2-grams in sequence), or the gross number of features (e.g. 10,000 as opposed 20,000) extracted from text. Considering the sensitive nature of most classifiers with regard to changes in training data, it is more than likely that their performance differs essentially with feature factors. After all, classifiers are trained on *features*, not training examples they are extracted from, and so the number and quality of features should matter more than number of examples as such. Some system factors are also still uncharted that relate to the details of its sense decision procedure. For instance, classifier parameters were shown by [3] to have considerable effect on performance, and the specifics of the WSD method itself will obviously have an effect (e.g. [2] showed that a different feature selection scheme shifts the classifier's strong region quite considerably).

Development of highly accurate best-system predictors depends on adequate accounting of all the factors in the WSD task setting. Once such accuracy is achieved, we can directly compare other systems to each other across datasets and ultimately represent the regions of all systems (regardless of dataset and language) in one series of word spaces. Such advances are in our mind feasible in the near future and would certainly further contribute to an understanding of 'WSD equation', i.e. the exact contribution of system factors and how a system's strength shifts if we alter classifier, feature pool or any of the specifics in its decision procedure. Word spaces can be used for numerically assessing base system strength and similarity and thereby selecting maximally complementary (i.e. strong but dissimilar) systems. With that, building optimal ensembles becomes greatly facilitated, saving on computing and analysis time.

Acknowledgments

Work of the first author is supported by Department of Translation Studies of Helsinki University, Finland, and Language Technology Doctorate School of Finland (KIT). Second author is financed by COMAS Doctorate School of Jyväskylä University, Finland, and Instituto Politecnico Nacional (IPN), Mexico. Work of the third author is supported by the IPN.

References

1. Edmonds, P., and Kilgarriff, A. Introduction to the Special Issue on evaluating word sense disambiguation programs. Journal of Natural Language Engineering 8(4) (2002).
2. Forman, G., and Cohen, I. Learning from Little: Comparison of Classifiers Given Little Training. *HYPERLINK "http://ecmlpkdd.isti.cnr.it/"* ECML'04 . 15th European Conference on Machine Learning and the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (2004)
3. Hoste, V., Hendrickx, I., Daelemans, W. and A. van den Bosch. Parameter optimization for machine-learning of word sense disambiguation. Journal of Natural Language Engineering, 8(4) (2002) 311-327.
4. Legrand, S., Pulido JGR. A Hybrid Approach to Word Sense Disambiguation: Neural Clustering with Class Labeling. Knowledge Discovery and Ontologies workshop at 15th European Conference on Machine Learning (ECML) (2004).
5. Luo, F., Khan, L., Bastani F., Yen I-L and Zhou, J. A dynamically growing self-organizing tree (DGSOT) for hierarchical clustering gene expression profiles, Bioinformatics 2004 20(16):2605-2617, Oxford University Press. (2004)
6. Mierswa, I. and Wurst, M., Klinkenberg, R., Scholz, M. and Euler, T. YALE: Rapid Prototyping for Complex Data Mining Tasks, in Proceedings of the 12th ACM SIGKDD (KDD-06) (2006)
7. Mihalcea, R. Word sense disambiguation with pattern learning and automatic feature selection. Journal of Natural Language Engineering, 8(4) (2002) 343-359.
8. Mihalcea, R., Kilgarriff, A. and Chklovski, T. The SENSEVAL-3 English lexical sample task. Proceedings of SENSEVAL-3 Workshop at ACL (2004).

9. Pedersen, T. Machine Learning with Lexical Features: The Duluth Approach to Senseval-2. Proceedings of SENSEVAL-2: Second International Workshop on Evaluating Word Sense Disambiguation Systems (2002).

10. Saarikoski, H. and Legrand, S. Building an Optimal WSD Ensemble Using Per-Word Selection of Best System. In CIARP-05, 11th Iberoamerican Congress on Pattern Recognition, Cancun, Mexico, to appear in Lecture Notes in Computer Science, Springer (2006).

11. Witten, I., Frank, E. Data Mining: Practical Machine Learning Tools and Techniques (Second Edition). Morgan Kaufmann
(2005).

12. Yarowsky, D., S. Cucerzan, R. Florian, C. Schafer and R. Wicentowski. The Johns Hopkins SENSEVAL2 System Descriptions. Proceedings of SENSEVAL-2 workshop (2002).

13. Yarowsky, D. and Florian, R. Evaluating sense disambiguation across diverse parameter spaces. Journal of Natural Language Engineering, 8(4) (2002) 293-311.

14. Zavrel, J., S. Degroeve, A. Kool, W. Daelemans, K. Jokinen. Diverse Classifiers for NLP Disambiguation Tasks. Comparisons, Optimization, Combination, and Evolution. TWLT 18. Learning to Behave. CEvoLE 2 (2000) 201–221.

14. Seo, H-C., Rim, H-C. and Kim, S-H. KUNLP system in Senseval-2. Proceedings of SENSEVAL-2 Workshop (2001) 222-225.

15. Strapparava, C., Gliozzo, A., and Giuliano, C. Pattern abstraction and term similarity for Word Sense Disambiguation: IRST at Senseval-3. In Proceedings of SENSEVAL-3 workshop (2004).

8. Manning, C., Tolga Ilhan, H., Kamvar, S., Klein, D. and Toutanova, K. Combining Heterogeneous Classifiers for Word-Sense Disambiguation. Proceedings of SENSEVAL-2, Second International Workshop on Evaluating WSD Systems (2001) 87-90.

5. Lee, Y-K., Ng, H-T., and Chia, T-K. Supervised Word Sense Disambiguation with Support Vector Machines and Multiple Knowledge Sources. In Proceedings of SENSEVAL-3 workshop (2004).

2. Grozea, C. Finding optimal parameter settings for high performance word sense disambiguation. In SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona, Spain (2004).