OE: WSD Using Optimal Ensembling (OE) Method

Harri M. T. Saarikoski
Helsinki University
Language Technology PhD Programme
F-00014 Helsinki, Finland
harri.saarikoski@helsinki.fi

Abstract

Optimal ensembling (OE) is a word sense disambiguation (WSD) method using word-specific training factors (average positive vs negative training per sense, *posex* and *negex*) to predict best system (classifier algorithm / applicable feature set) for given target word. Our official entry (OE1) in Senseval-4 Task 17 (coarse-grained English lexical sample task) contained many design flaws and thus failed to show the whole potential of the method, finishing -4.9% behind top system (+0.5 gain over best base system). A fixed system (OE2) finished only -3.4% behind (+2.0% net gain). All our systems were 'closed', i.e. used the official training data only (average 56 training examples per each sense). We also show that the official evaluation measure tends to favor systems that do well with high-trained words.

1 Introduction

Optimal ensembling is a novel method for combining WSD systems and obtaining higher classification accuracy (presented more fully in Saarikoski et al. 2007). The essential difference from other ensembling methods (such as various types of voting ensembles and cross-validation based best machine selection) is that best machine is predicted using factors calculated from words (e.g. number of senses) and their training data (e.g. number of training examples per sense). The method is loosely based on findings of system performance differences in both WSD (different machines by Yarowsky et al., 2002 and different feature sets by Mihalcea, 2002) and other classification tasks such as text categorization (Forman et al., 2004, Bay et al., 2002).

2 Method

We first describe in detail the two selection routines in OE as deployed in this experiment.

2.1 Machine (Mach) Selection

We selected support vector machine (SVM) (Vapnik, 1995) and Naive Bayes (NB) (John et al. 1995) as classifiers for our base systems to be optimally ensembled. This was mainly because of their attested strength at earlier Senseval evaluations (Edmonds et al. 2002, Mihalcea et al. 2004) and mutual complementarity discovered by us (Saarikoski et al., 2007). Original batch of candidate machines that we tested for OE using Senseval-2 dataset included the following classifiers: Decision Stump, Decision Tree with various values of confidence (c) parameter 0.05, 0.15, 0.25 and instance-based classifier with *k* values ranging from 1..15 at intervals of two [1]. After cross-validation runs against current dataset (see below), however, SVM and NB proved again to be overall strongest regardless of training input, so we built OE around those two classifiers.

2.2 Feature Set (Fset) Selection

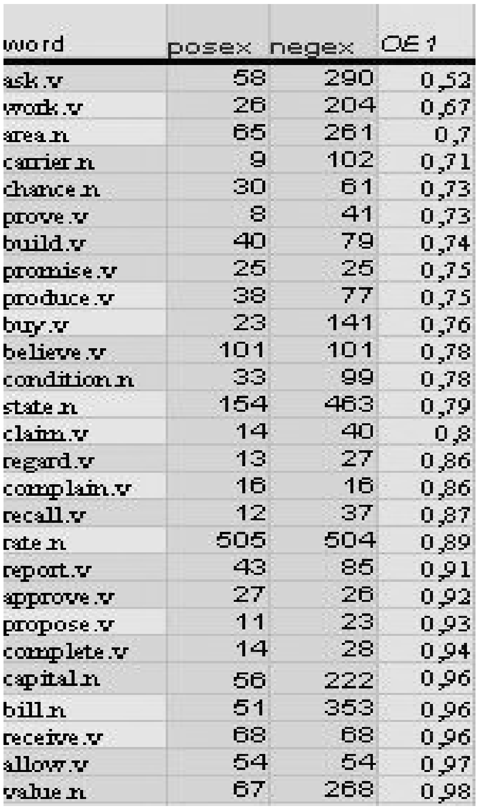
We extracted three contextual feature sets from training data for all words to train the machines: 1-grams (1g) and sequential 2-grams both from whole instance (2g) as well as part-of-speech tags from local 1-word window around and including target word (pos3). We also used three 'multifsets' (1g-2g, 1g-pos3, 2g-pos3).

[1] We used Weka implementations (J48, Ibk, SMO, Decision Stump, NaiveBayes) of these algorithms (Witten, 2005).

Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007), pages 257–260,
Prague, June 2007. ©2007 Association for Computational Linguistics

2.3 Best-System Prediction Factors

In Figure 1, we quote prediction factors used for predicting best system for some test words.



word	posex	negex	OE1
ask.v	58	290	0,52
work.v	26	204	0,67
area.n	65	261	0,7
carrier.n	9	102	0,71
chance.n	30	61	0,73
prove.v	8	41	0,73
build.v	40	79	0,74
promise.v	25	25	0,75
produce.v	38	77	0,75
buy.v	23	141	0,76
believe.v	101	101	0,78
condition.n	33	99	0,78
state.n	154	463	0,79
claim.v	14	40	0,8
regard.v	13	27	0,86
complain.v	16	16	0,86
recall.v	12	37	0,87
rate.n	505	504	0,89
report.v	43	85	0,91
approve.v	27	26	0,92
propose.v	11	23	0,93
complete.v	14	28	0,94
capital.n	56	222	0,96
bill.n	51	353	0,96
receive.v	68	68	0,96
allow.v	54	54	0,97
value.n	67	268	0,98

Figure 1. Prediction factors and OE1 accuracy for some test words in Senseval-4 Task 17 (sorted by OE1 accuracy at the word).

3 System Descriptions

We designed and ran two systems:

OE1 (official): For OE1, we used two machines in three configurations (SVMc=0.1, SVMc=1.0, NB) trained on 3 feature sets, totalling at 3*3 = 9 base systems (number of machines * number of fsets for each). Selection of *c(omplexity)* parameter for SVM was based on previous knowledge of performance differences of c=0.1 and c=1.0 based systems as reported in Saarikoski et al. (2007). This is based on accounts by e.g. Vapnik (1995) that lower *c* value makes the classifier generate a more complex training model which is more suitable for tougher words (lower posex, higher negex).

We learned the best-system predictor model using performance data from Senseval-4 10CV runs only. For 70 words where two fsets performed within +/-5% of each other, we added the next best fset into a 'multifset'.

OE2 (unofficial): This system incorporated the following fixes to OE1 (see Discussion below for motivations for these fixes): First, we significantly reduced the base system grain. We only used two machines strongest in 10CV runs (SVMc=0.1 and NB) and these machines were trained with fsets found best for those machines in 10CV runs: pos3 for both machines, SVMc=0.1 was additionally trained with 1g and NB with 2g respectively. This resulted in a 2 * 2 = 4-system ensemble. Best fset was still selected on the basis of 10CV runs.

As training data for the best-machine predictor, we used the performance profiles of about 50 systems (both our own and Senseval systems) run mainly against Senseval-2 English lexical sample dataset. We decided to use only two prediction factors (posex and negex, see Figure 1) to predict best machine for each word. This was because previously we had found these two machines (SVM and NB) particularly differing with regard to the combination or cross-section of these two factors. (For illustration of the predictor model with posex and negex as the two axes and discussion of other possible factors, see Saarikoski et. al, 2007. As to reasons for such a performance difference between any two classification machines, see also Yarowsky et al., 2002).

Difference in the best-system predictions of these two systems (OE1 vs OE2) was substantial: 33 words fully changed machine (from SVM to NB or vice versa), 40 words partially changed the system (change of SVM configuration or change of fset from multifset to single fset). Only 27 words kept the same machine in same configuration and fset. We can therefore call OE2 a substantial revision of OE1 (in effect a rather total departure from CV-based selection toward actual word factor based optimal ensembling).

In both OEs, the mach-fset combination predicted to be the best for a word was run against the test instances of that word [2]. In case of 'multifsets', each single fset had equal probability-based vote in disambiguating the test instances of

[2] SyntaLex code (Mohammad and Pedersen, 2002, http://www.d.umn.edu/~tpederse/syntalex.html) was used for extracting n-grams and carrying out disambiguation. Brill Tagger (Brill, 1995) was used for extracting PoS tags. Weka library of classifiers (Witten, 2005) was used to run cross-validations and best-system predictors.

that word. As usual, the sense with highest probability was chosen as answer for each instance.

4 Test Results

Here are the results:

system name	gross gain	net gain	accuracy[3]
OE1	**+3.0 (+7.8)**	**+0.5 (+4.4)**	**83.8**
OE2	**+2.3 (+7.0)**	**+2.0 (+5.8)**	**85.3**

Table 1. Results of OE systems. In columns 2-3, macro (micro) averaged per-word gross and net gains calculated from actual test runs (not 10CV runs) are reported. Column 4 reports the official macro-averaged accuracy for all words of our systems. (Differences of the respective benefits of these evaluation measures are outlined in Discussion below and more generally in Sebastiani (2002). Terms 'gross (or potential) gain' and 'net (realized) gain' are defined in Saarikoski et al. (2007).).

5 Discussion

We now turn to analyze these results. We can first note that results are largely in line with our previous findings with OEs and other types of ensembles (see Saarikoski et al., 2007). In what follows we attempt to account for the results: why OE1 finished as much behind top system and also why OE2 performed that much better than OE1. This first 'known issue' concerns both OEs:

(1) Base system accuracy was low because we did not use strong fsets: Our official entry finished at 7th place in the evaluation, -4.9% behind top system while the inofficial entry would have finished in 5th place (-3.4% behind). We attribute this mainly to the absence of more advanced feature sets. For example, we did not employ syntactic parse features (such as predicate-object pairs) from which Yarowsky et al. (2002) showed +2% gain. We would also naturally lose to any systems using extra training or lexical knowledge (e.g. 2nd place finisher UBC-ALM, at 86.9 accuracy, used both semantic domains and SemCor corpus). But without knowing how much extra knowledge such 'open' systems used, we cannot say by how much.

Specifically in OE1 entry, there were two basic design flaws which we address next.

(2) Base system grain was too high to produce enough net gain: The base system grain (18 base systems) we attempted to predict in OE1 was far too great since prediction accuracy rapidly decreases when adding new systems. The grain was also unnecessarily great, since the 4-grain we used for OE2 could harvest most of the gross gain (cf. gross gains of the two systems in Table 1).

(3) Using 10CV runs uncritically for best fset selection: This was ill-advised because of many reasons. First, selecting best fset for WSD based on CV runs is known to be a difficult task (Mihalcea, 2002). Prediction accuracy for the three fsets we used for OE1 was 0.74, i.e. for 26 words out of 100 best fset was mispredicted. About half of these were cases where machine was mispredicted as well and average loss tended to be even greater. Second, multifsets could not be 10CV-tested with the Weka machine-learning toolkit we used (Witten, 2005). Our custom resolution to this multifset selection task was to select best and next best fset. This turned out to produce many false predictions, some of which were quite substantial (> 10% loss to best fset). For instance, at *system.n* we lost > 30% from selecting NB-2g instead of actual best system (NB-pos3). Third, only after submitting the entry, we also realized two strongest fsets are not necessarily complementary (i.e. that each would contain relevant clues for *different* test instances) and that learning machines might be confused (i.e. could not effectively carry out feature selection and weighting) by the profusion and heterogeneity of features in multifsets. In fact, we found that omitting multifsets from OE1 (i.e. having 3 single fsets with the same 3 machines = 6-system OE) would have worked slightly better than OE1 (3*3=9): the accuracy rose from 83.8 to 84.1. Fourth, it was found previously (Saarikoski et al., 2007) that CV-based best system prediction scheme tends to produce less gain than OE (cf. accuracy of OE1 < OE2 in Table 1).

The remaining argument discusses Senseval evaluation measure (applies to all OE systems):

(4) Official evaluation measure is particularly unfavorable to OE systems: Senseval scoring scheme[4] is calculated as the number of instances disambiguated correctly divided by number of all

[3] Best base system in both OEs was NB-pos3 (83.3).

[4] Documentation for scoring scheme can be found at: http://www.cse.unt.edu/~rada/senseval/senseval3/scoring/

instances in test dataset. This measure (termed 'macro-averaged accuracy' in Sebastiani, 2002) is known to upweigh classification cases (words) that have more test instances. While we recognize the usefulness of this measure, we calculated in Table 1 the alternative measure (termed 'micro-averaged accuracy' in Sebastiani, 2002). It differs from the former (defined by e.g. Sebastiani, 2002) in that *all words are treated equally* (i.e. 'normalized') regardless of number of test instances. In addition, it has been Senseval practice (Edmonds et al. 2002, Mihalcea et al. 2004) that words with great number of test instances tend to have an equally great number of training instances. At such 'easier' words, system performance differences (sysdiff) occur much less and since OE is based on locating and making use of sysdiff, it cannot perform well. Therefore, it is liable to lose to single-machine systems with inherently stronger fsets (see point 1 above). For these reasons, the measures are very different with the latter revealing the OE potential more appropriately.

In fact, we estimate that only 40 out of the 100 test words in this dataset show any kind of sysdiff between most participating systems (> 5% macro-averaged sysdiff per word). Furthermore, only 20 of them only are likely to produce substantial sysdiff (> 10%). For example, in our 10CV runs, we got 0.99 accuracies by all base systems for the very highly trained word *say.v* with posex > 500. If there was a participating system that achieved 1.00 in such a single high-train word (*say.v*), the huge number of test instances of that word raised its macro-averaged accuracy, winning considerably over systems performing well with low-train words (e.g. *propose.v* with posex=11 and negex=24 and grain=3 where both OE1 and OE2 performed at 0.93 accuracy owing to correct best system choice). In other words, the official measure does not account for the finding (Yarowsky et al., 2002 and Saarikoski et al., 2007) that systems considerably differ precisely in terms of their ability to disambiguate high/low-train words (measured by posex/negex factors). Therefore, it can be said that the official measure fails to treat all systems equally.

6 Conclusion and Further Work

Since OE is a generic method that can be applied to any base systems, we believe it has a place in WSD methodology. With remaining open questions resolved (optimizing system grain to feasible prediction accuracy, discovering more predictive factors for both machines and fsets, understanding how the evaluation measures complete each other), it is probable that OE can improve current state of the art WSD systems (especially if provided with stronger while still complementary base systems). Though OE systems run the risk that OE may in fact be inferior to its best base system, we would like to note that thus far no OE of ours (around 10-15 different tests) has failed to produce net gain.

References

Bay, S. D., and Pazzani, M. J. Characterizing model errors and differences. In 17th International Conference on Machine Learning (2000)

Brill, E. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging Computational Linguistics (1995)

Edmonds, P., and Kilgarriff, A. Introduction to the Special Issue on evaluating word sense disambiguation programs. Journal of Natural Language Engineering 8(4) (2002)

Forman, G., and Cohen, I. Learning from Little: Comparison of Classifiers Given Little Training. In ECML, 15th European Conference on Machine Learning and the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases (2004)

John, G. and Langley, P. Estimating Continuous Distributions in Bayesian Classifiers. Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann, San Mateo (1995)

Mihalcea, R. Word sense disambiguation with pattern learning and automatic feature selection. Journal of Natural Language Engineering, 8(4) (2002)

Mihalcea, R., Kilgarriff, A. and Chklovski, T. The SENSEVAL-3 English lexical sample task. Proceedings of SENSEVAL-3 Workshop at ACL (2004)

Mohammad, S. and Pedersen, T (2004). Complementarity of Lexical and Simple Syntactic Features: The Syntalex Approach to Senseval-3. Proceedings of Senseval-3

Saarikoski, H., Legrand, S., Gelbukh, A. (2007) Case-Sensitivity of Classifiers for WSD: Complex Systems Disambiguate Tough Words Better. In CICLING 2007 and Lecture Notes in Computer Science, Springer

Sebastiani, F. Machine learning in automated text categorization, ACM Computing Surveys (CSUR), Vol. 34 Issue 1 (2002) ACM Press, New York, NY, USA.

Vapnik, V. N. The Nature of Statistical Learning Theory. Springer (1995)

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

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