Natural Language Engineering **8** (4): 293–310.
DOI: 10.1017/S135132490200298X Printed in the United Kingdom

Evaluating sense disambiguation across diverse parameter spaces

DAVID YAROWSKY and RADU FLORIAN

Department of Computer Science and Center for Language and Speech Processing, Johns Hopkins University, MD 21218, USA
e-mail: {yarowsky,rflorian}@cs.jhu.edu

(*Received 19 November 2001; revised 20 June 2002*)

Abstract

This paper presents a comprehensive empirical exploration and evaluation of a diverse range of data characteristics which influence word sense disambiguation performance. It focuses on a set of six core supervised algorithms, including three variants of Bayesian classifiers, a cosine model, non-hierarchical decision lists, and an extension of the transformation-based learning model. Performance is investigated in detail with respect to the following parameters: (a) target language (English, Spanish, Swedish and Basque); (b) part of speech; (c) sense granularity; (d) inclusion and exclusion of major feature classes; (e) variable context width (further broken down by part-of-speech of keyword); (f) number of training examples; (g) baseline probability of the most likely sense; (h) sense distributional entropy; (i) number of senses per keyword; (j) divergence between training and test data; (k) degree of (artificially introduced) noise in the training data; (l) the effectiveness of an algorithm's confidence rankings; and (m) a full keyword breakdown of the performance of each algorithm. The paper concludes with a brief analysis of similarities, differences, strengths and weaknesses of the algorithms and a hierarchical clustering of these algorithms based on agreement of sense classification behavior. Collectively, the paper constitutes the most comprehensive survey of evaluation measures and tests yet applied to sense disambiguation algorithms. And it does so over a diverse range of supervised algorithms, languages and parameter spaces in single unified experimental framework.

1 Prior comparative surveys of sense disambiguation performance

Most previous comparative surveys of sense disambiguation performance have been limited to a single algorithm, a single language, a single word (or a few words) or all of the above. For example, Gale, Church and Yarowsky (1992), in one of the most comprehensive parameter-based studies to-date (including training size, context width and introduced noise), were limited to a single algorithm (a ratio-based Bayesian classifier) over binary sense distinctions on 12 English words. Yarowsky (1993) further detailed the contributions of variable feature types and part-of-speech sensitivity to context width, but was limited to one algorithm (decision list) and 30 binary homographs. In contrast, Leacock, Towell and Voorhees (1993) compared three distinct algorithms (a content vector model, a Bayesian classifier, and a single-layer neural-net) on a more heavily polysemous word (*line*) but varied only training

data size. Mooney (1996) extended the *line*-based comparative survey to algorithms including Naïve Bayes, 3-nearest neighbors, perceptron, decision tree, decision list and PFOIL inductive logic programming variants. However, Mooney restricted comparisons to training size, training time and testing time. His conclusions that Naïve Bayes was consistently the top performer in this set of algorithms finds some support in the results of the current study below. In more recent comparative surveys, the inclusion of learning curves (based on variable training data size) have become the standard, but additional new parameters are rarely explored. One addition includes Ng's (1997) comparison of performance across different corpora (Brown Corpus and *Wall Street Journal*). Other large-scale comparative studies include Pedersen (2001), who compared Bayesian and decision-tree algorithms in detail, and Stevenson and Wilks (2001), who investigated the relative efficacy of such knowledge sources as LDOCE subject codes and selectional preference over major parts of speech and sense granularities.

A limiting factor in prior comparative analyses was the lack of a large-scale standard evaluation set. SENSEVAL1 (Kilgarriff and Palmer 2000) dramatically improved that status quo, and for the first time over 18 sense disambiguation systems were compared in a common evaluation framework. Kilgariff and Rosenzweig (2000) rigorously contrasted systems by keyword part-of-speech, presence of supervision, number of senses and sense entropy. However, because each system was developed and executed at independent sites, the evaluation process could not modify and contrast system internal parameters (such as variable context width or inclusion of feature classes) or even variable properties of data sets (such as training set size). Furthermore, systems not only used different algorithms but also different feature representations and feature extraction quality. Thus, it was not possible in this heterogeneous survey to isolate differences due to algorithm design, feature space utilized or the parameter settings chosen. While such a multi-site comparative exercise has been invaluable for exploring unprecedented diversity of methods in a common test set and standards, there remains a need for single-site comparative studies more closely able to control and systematically vary internal and external parameters across algorithms, maintaining other variables such as the utilized feature space as constant as possible. This paper presents such a survey.

2 Experimental framework and utilized algorithms

All the algorithms contrasted in this study are based on a shared, uniform and rich contextual feature space, including word, part-of-speech and lemma in both variable-width bag-of-words wide context and in local n-gram collocations. They also include feature associations in salient syntactic relationships, such as verb-object, noun-modifier, etc. The feature space is shared with the classifier combination study in Florian, Cucerzan, Schafer and Yarowsky (this issue) and described more fully there. The SENSEVAL2 WSD training data (Edmonds and Cotton 2001) are used with five-fold cross-validation for all evaluation. Details including training data size and number of senses of all 73 English polysemous keywords, can be found in Table 1 (Appendix).

Table 1. *Keyword-itemized performance on* SENSEVAL2 *English lexical sample task*

Model	Num Samples	Num Senses	ML	Entr	FENBayes	BayesRatio	Cosine	DL	TBL
begin.v	557	8	59.1%	0.2	79.4%	79.2%	80.3%	81.3%	**83.1%**
call.v	132	23	25.7%	0.5	**43.9%**	38.6%	35.6%	39.4%	40.2%
carry.v	132	27	23.5%	0.6	37.9%	**43.2%**	43.2%	39.4%	40.1%
collaborate.v	57	2	91.2%	0.1	86.1%	**94.7%**	87.9%	91.2%	**94.7%**
develop.v	133	15	30.1%	0.5	36.9%	38.4%	**41.3%**	40.6%	36.0%
draw.v	82	32	8.5%	0.7	30.4%	31.6%	32.9%	**35.2%**	26.9%
dress.v	119	14	39.3%	0.4	**60.5%**	59.6%	53.8%	45.4%	56.2%
drift.v	63	9	20.5%	0.5	37.9%	34.6%	28.3%	**41.2%**	33.3%
drive.v	84	15	32.1%	0.5	60.6%	60.7%	**61.8%**	54.7%	52.4%
face.v	186	7	83.3%	0.2	80.1%	79.0%	75.3%	**85.5%**	81.7%
ferret.v	2	**1**	**100.0%**	0.0	**100.0%**	**100.0%**	**100.0%**	**100.0%**	**100.0%**
find.v	132	17	15.9%	0.6	**41.6%**	35.6%	35.6%	34.8%	28.7%
keep.v	133	27	36.9%	0.5	35.3%	44.4%	36.9%	52.7%	**60.9%**
leave.v	132	14	28.9%	0.5	**44.8%**	43.9%	41.0%	43.3%	39.5%
live.v	129	10	49.6%	0.4	**64.2%**	62.7%	61.1%	61.9%	62.7%
match.v	86	8	36.1%	0.4	36.0%	30.1%	33.7%	33.5%	**45.4%**
play.v	129	25	10.8%	0.5	44.9%	40.3%	37.9%	**45.7%**	44.1%
pull.v	122	33	22.2%	0.6	**48.4%**	42.7%	47.7%	44.4%	44.4%
replace.v	86	4	51.2%	0.3	45.4%	45.3%	44.2%	47.7%	**61.8%**
see.v	131	21	31.3%	0.5	**37.4%**	36.6%	27.4%	32.1%	32.8%
serve.v	100	12	26.0%	0.5	**59.0%**	54.0%	53.0%	47.0%	50.0%
strike.v	104	26	9.6%	0.6	**43.1%**	40.3%	34.5%	40.2%	**43.1%**
train.v	125	9	23.2%	0.4	55.2%	48.8%	44.0%	**56.8%**	45.6%
treat.v	88	6	29.5%	0.3	52.4%	53.4%	46.6%	43.2%	**55.6%**
turn.v	131	43	9.9%	0.7	53.4%	52.7%	54.2%	55.7%	**61.0%**
use.v	147	7	68.0%	0.2	66.6%	65.9%	51.0%	**70.0%**	70.0%
wander.v	100	4	**83.0%**	0.1	78.0%	79.0%	63.0%	81.0%	82.0%
wash.v	25	13	8.0%	0.8	52.0%	52.0%	56.0%	**68.0%**	40.0%
work.v	119	21	27.6%	0.5	44.6%	**46.3%**	40.4%	40.5%	39.6%
art.n	196	19	38.2%	0.4	59.7%	65.9%	63.8%	61.7%	**67.3%**
authority.n	184	11	33.7%	0.3	**69.1%**	69.0%	64.1%	60.4%	66.4%
bar.n	304	22	41.8%	0.4	**71.4%**	71.0%	69.4%	63.1%	65.1%
bum.n	92	6	70.6%	0.3	69.5%	70.6%	62.0%	71.8%	**73.9%**
chair.n	138	8	82.6%	0.2	**91.3%**	91.3%	88.4%	89.9%	88.4%
channel.n	145	10	40.7%	0.4	60.0%	**62.1%**	61.4%	49.7%	48.3%
child.n	129	9	60.4%	0.2	68.2%	66.6%	64.3%	72.1%	**78.2%**
church.n	128	7	56.4%	0.2	**75.9%**	72.7%	65.6%	62.6%	68.9%
circuit.n	170	16	27.1%	0.5	**83.5%**	**83.5%**	71.8%	63.5%	62.4%
day.n	289	18	60.6%	0.3	69.9%	**72.7%**	67.5%	70.6%	72.3%
detention.n	63	6	72.9%	0.3	**96.9%**	**96.9%**	90.4%	**96.9%**	**96.9%**
dyke.n	58	4	83.0%	0.3	81.5%	79.8%	71.4%	71.1%	**84.7%**
facility.n	114	6	53.5%	0.3	**77.9%**	70.1%	70.2%	72.0%	73.7%
fatigue.n	85	8	71.8%	0.3	87.1%	85.9%	**88.2%**	84.7%	**88.2%**
feeling.n	102	5	64.7%	0.2	68.7%	**72.6%**	69.7%	62.8%	69.7%
grip.n	102	7	53.8%	0.3	**69.8%**	63.9%	59.9%	58.8%	58.7%
hearth.n	64	5	**63.8%**	0.3	**63.8%**	60.8%	62.3%	63.8%	62.3%
holiday.n	62	8	88.8%	0.2	**96.9%**	**96.9%**	**96.9%**	95.3%	95.3%
lady.n	105	10	69.5%	0.3	**81.9%**	**81.9%**	79.0%	79.0%	78.1%
material.n	140	17	37.1%	0.4	63.6%	**64.3%**	61.4%	49.3%	54.3%
mouth.n	119	12	50.4%	0.3	59.7%	60.5%	58.9%	54.7%	**63.2%**
nation.n	75	5	**85.3%**	0.2	81.3%	81.3%	74.7%	81.3%	82.7%
nature.n	92	9	48.9%	0.4	60.9%	**65.3%**	64.2%	65.1%	58.7%

Table 1. (*Cont.*)

Model	Num Samples	Num Senses	ML	Entr	FENBayes	BayesRatio	Cosine	DL	TBL
post.n	157	15	40.8%	0.4	69.4%	**73.3%**	72.0%	64.3%	70.1%
restraint.n	91	9	35.1%	0.4	69.3%	**75.8%**	67.0%	57.2%	61.6%
sense.n	107	9	35.7%	0.4	68.3%	71.9%	72.7%	66.5%	**77.5%**
spade.n	65	8	67.7%	0.3	83.1%	**84.6%**	81.5%	76.9%	81.5%
stress.n	79	7	57.0%	0.3	63.3%	67.2%	**73.5%**	63.3%	67.1%
yew.n	57	4	85.9%	0.2	**94.5%**	90.9%	**94.5%**	92.7%	91.1%
blind.a	108	9	62.9%	0.3	**74.2%**	71.5%	70.5%	72.3%	72.2%
colourless.a	68	3	77.9%	0.2	80.9%	**82.4%**	82.3%	77.8%	81.0%
cool.a	106	8	50.0%	0.4	**70.7%**	59.4%	52.9%	66.1%	56.6%
faithful.a	47	3	72.2%	0.2	65.6%	67.8%	59.6%	**74.2%**	70.0%
fine.a	142	13	40.7%	0.4	58.4%	**64.1%**	64.1%	55.7%	61.9%
fit.a	57	4	63.3%	0.2	83.9%	**91.1%**	76.8%	89.1%	**91.1%**
free.a	165	19	49.7%	0.3	**71.5%**	67.9%	68.5%	70.3%	70.3%
graceful.a	56	2	85.6%	0.1	83.8%	85.6%	85.6%	82.3%	**87.4%**
green.a	190	19	75.8%	0.3	**84.7%**	82.1%	74.2%	80.5%	83.2%
local.a	75	3	68.0%	0.2	72.0%	**90.7%**	78.7%	89.3%	89.3%
natural.a	206	25	31.0%	0.5	62.6%	**63.1%**	56.3%	52.9%	51.4%
oblique.a	57	4	57.6%	0.3	70.3%	72.3%	73.9%	**77.0%**	73.6%
simple.a	130	6	50.0%	0.3	45.4%	49.2%	50.0%	**54.6%**	52.3%
solemn.a	52	2	**90.5%**	0.1	88.5%	88.5%	78.5%	**90.5%**	88.5%
vital.a	74	8	86.5%	0.2	86.4%	89.1%	86.4%	89.1%	**90.6%**
Overall Mean	167.2	13.4	48.3%	0.4	65.2%	65.2%	62.2%	63.0%	64.5%
No times was Max			5		25	21	7	13	20

The algorithm set investigated here includes a standard cosine vector model, non-hierarchical decision lists (DL; as described in Yarowsky (1996)), and a variant of Transformation-Based Learning (TBL) optimized for word-sense disambiguation (Florian *et al*., this issue). It also includes the Naïve Bayes model and a variant BayesRatio (BR) model, using the $\frac{P(s|d)}{P(\neg s|d)}$ likelihood ratio model described in Gale *et al.* (1992). All of these algorithms utilize the same rich feature space to facilitate direct comparison.[1] The selective inclusion and omission of these features is systematically explored in section 3.2. As an additional benchmark, the traditional Naïve Bayes using only unordered bag-of-words features is directly compared in certain experiments as a stand-alone algorithm. Thus for clarity, the naïve Bayes model with its feature space significantly augmented with this study's full set of position-sensitive, syntactic and local collocation features, is henceforth referred to as FENBayes (Feature-Enhanced Naïve Bayes).

Finally, based both on similarities in algorithm design and empirical behavior detailed in section 3.7, we will classify these algorithms as one of two major types:

[1] Positionally sensitive features are easily added to the traditional bag-of-words models such as cosine by simply subscripting vector tokens with their feature class. This distinguishes the keyword-adjacent *high_L church* from an unsubscripted *high* in position-independent bag-of-words context.

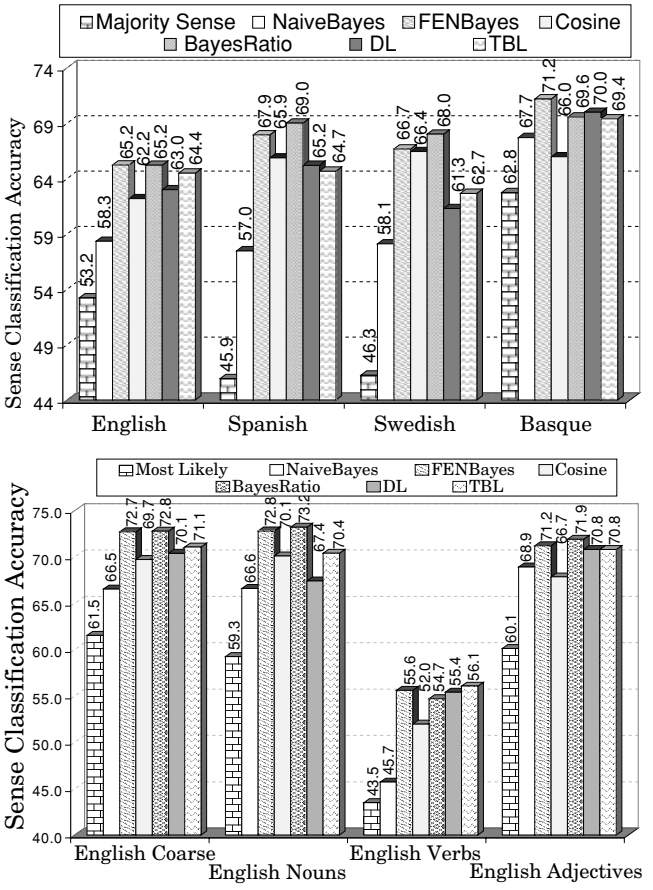


Fig. 1. Performance based on language, sense granularity and part-of-speech.

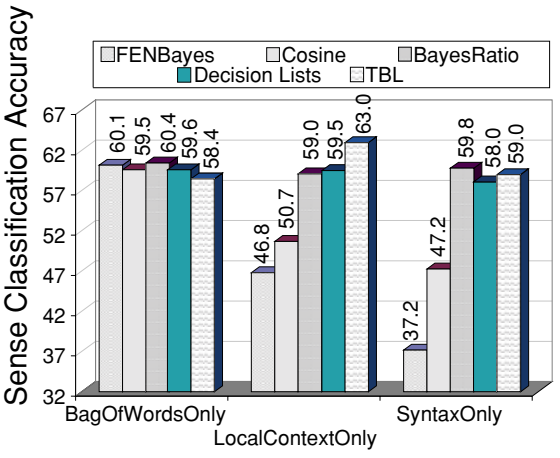
discriminative (DL and TBL) or *aggregative* (cosine, FENBayes and BR). The aggregative models integrate *all* available evidence in favor of a sense and then select the sense with the maximum cumulative support, while the discriminative models tend to rely on one or a few features in any given context that most efficiently partition or discriminate the candidate sense space.

3 Comparative study of algorithms and diverse parameter spaces

The following seven subsections provide a comprehensive and systematic study of the effects of parameter-space variation over the six focus algorithms outlined in section 2. Performance on the 73 English keywords is further detailed in Table 1.

3.1 Performance sensitivity to language and part-of-speech

Figure 1 compares performance of the algorithms by language and part-of-speech. FENBayes and BayesRatio are consistently the top performing methods. One exception is the Decision List classifier, which achieves second place on Basque.



Accuracy drop relative to full system					
Features used	Aggregative			Discrimin.	
	FENB	CSN	BR	TBL	DL
Omit Bag-of-words Ftrs	−14.7	−8.1	−5.3	−0.5	−2.0
Omit Local Collocations	−3.5	−0.8	−2.2	−2.9	−4.5
Omit Syntactic Features	−1.1	−0.8	−1.3	−1.0	−2.3
Bag-of-words Ftrs *Only*	−6.4	−4.8	−4.8	−6.0	−3.2
Local Collocations *Only*	−18.4	−11.5	−6.1	−1.5	−3.3
Syntactic Features *Only*	−28.1	−14.9	−5.4	−5.4	−4.8

Fig. 2. English performance based on variable exclusion and inclusion of feature types.

Cosine is on average weakest, especially on English and Basque. Also, on Spanish and Swedish the best aggregative models perform significantly[2] better a ($p \leq 10^{-4}$) than the discriminative ones, perhaps due to the less-frequent majority sense baseline observed in these languages.

Decision lists and TBL perform significantly lower on nouns ($p \leq .01$) than the best aggregative models, consistent with the observation that will be detailed in Section 3.3 that they are less effective in modeling multiple features in wide context. In contrast, their accuracy is closely competitive on verbs and adjectives, where single features tend to be independently predictive.

3.2 Performance sensitivity to feature type

This section explores both the relative contributions of individual feature types and the relative effectiveness of different algorithms in exploiting the individual feature types. Henceforth all results will be based on SENSEVAL2 English data.

Figure 2 illustrates the performance differences between aggregative and discriminative models based on available feature type. Figure 2(a) shows the performance

[2] All significance tests in this article were performed using the paired McNemar test.

of individual models when only provided with features of three types.[3] Figure 2(b) details the observed loss in performance when either *only using* one of the positional feature classes or *excluding* one of the positional feature classes relative to the algorithm's full performance using all available feature types.

Figure 2(b) shows that the discriminative algorithms (TBL, DL) are generally effective at modeling each of these three positional feature types in isolation. As they tend to base their classifications on the single most reliable contextual feature (for DL) or few features (for TBL), restriction to the most information-bearing syntactic positions or local collocations does not greatly impoverish the feature space relative to the one they would utilize without restriction. In particular, decision lists tend to be most balanced in the relative contribution of individual feature classes; exclusion of any one feature type drops overall performance by a fairly uniform 2.3–4.5%, while performance using each feature class in isolation is nearly identical. The divergence is somewhat greater for TBL, which depends most heavily upon local context (2.9% loss when excluded and only 1.5% loss when used in isolation) and derives relatively little contribution from wider-context bag-of-words (only 0.5% drop when used in isolation, 6% drop when used exclusively).

In striking contrast, the aggregative models (Cosine and Bayes variants) depend heavily on the multiple reinforcing feature clues obtainable from wide context, and are severely hobbled by restriction to the few instantiated local features despite their relatively greater information content per feature. Excluding wide-context bag-of-words features drops Bayes and Cosine performance by 15% and 8%, respectively, while using local context in isolation leads to a 18% and 11% drop, exactly the converse of TBL's behavior. This complementarity of expertise provides strong motivation for the productive classifier combination of these model variants described in Florian *et al*. (this issue).

Overall, the marginal performance improvement due to local-context collocational features is on average 3.3% (4.5% for decision lists). Syntactic positional features on average offer a weaker average marginal performance improvement of 1.4% (2.3% for decision lists), perhaps due to their sparser instantiation rate (29.4%) and the greater noise in syntactic dependency detection.

Figure 3 further details the contribution of syntactic features by part of speech and argument type. Nouns derive relatively little marginal benefit from syntactic features over simple local collocations (such as the words to their right and left) and wide context, while verbs (and to a lesser extent adjectives) derive greater marginal benefit from a more precise extraction of their syntactic arguments. As would be

[3] These feature types contrasted in Figure 2 are: (a) BagOfWords Context only (context modeled as a bag of contextual features undifferentiated by position) although utilizing variable context widths optimized on a held-out development data, and including all feature types (such as lemmas) not just raw words, (b) Local Context only, which includes only features in bigrams and trigrams immediately adjacent to the keyword, and (c) Syntactic Features Only, which includes those features in specific syntactic relationships to the keyword such as VerbObject and NounModifier. The latter suffer from sparse or incomplete data problems because many of these syntactic relationships do not apply to individual keywords contexts, and those that do (such as the verb in a subject-predicate relationship) might have been unreliably identified due to syntactic ambiguity.

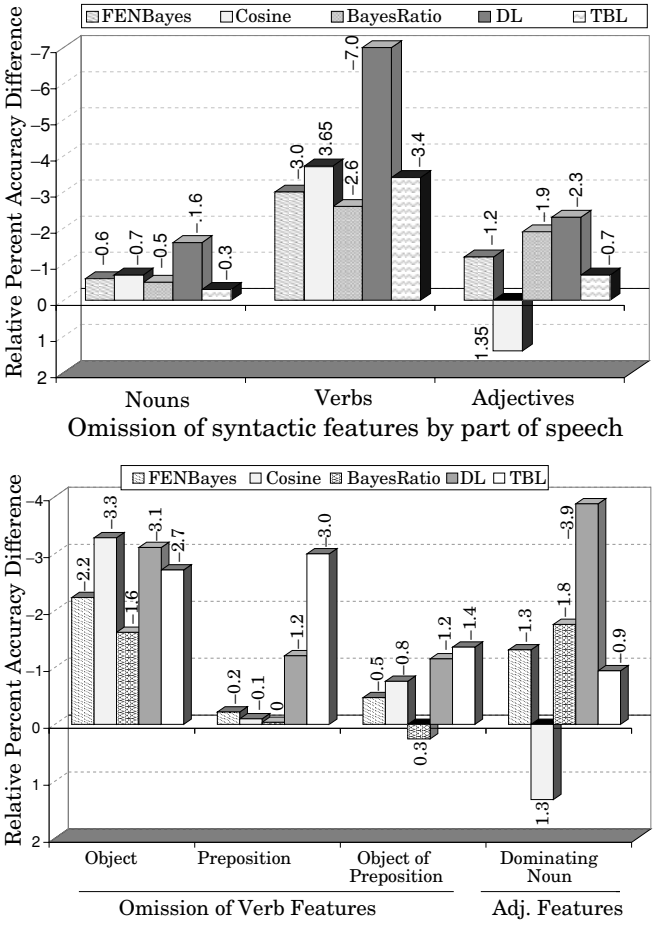


Fig. 3. The contributions of various types of syntactic features as measured by the relative system accuracy loss on their omission.

expected, verbs derive the greatest marginal benefit from extracting their object, although the discriminative models (DL and TBL) are also effective at utilizing any preposition or object-of-preposition arguments when present. Decision lists also benefit strongly from more precise isolation of an adjective's dominating noun, such as in a copular relationship distinct from simple adjacent collocation. These results may be specific to English, however, and one would expect a greater contribution of syntactic analysis to languages where the key arguments appear with freer word order or in less typically adjacent positions than in English.

3.3 Performance sensitivity to context window size

There are striking differences in model performance based on utilized wide-context window size. Figure 4(a) further substantiates that TBL and decision lists are the most effective at exploiting narrow context window sizes, while the performance of the aggregative models continues to grow as more potential reinforcing features

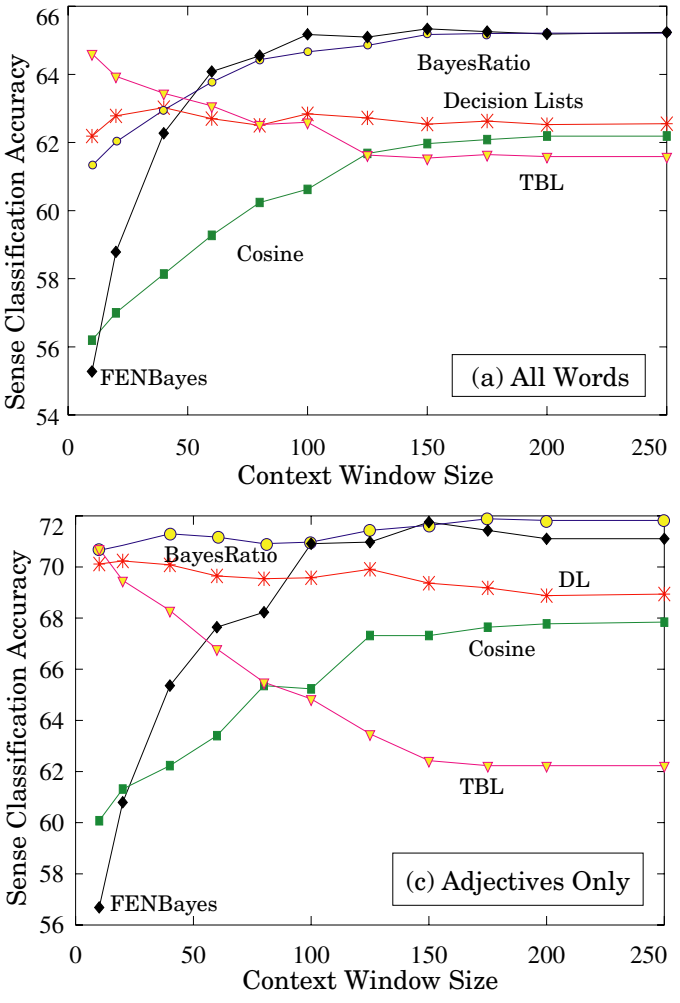


Fig. 4. Classification accuracy based on variable context width.

are made available to them. Interestingly, TBL's performance actually decreases as available context width grows, indicating that spurious associations from wide context may be overwhelming local contextual signal. This motivates the selection of a relatively narrow ± 10 word maximum context window for TBL based on held-out development data parameter estimation. Decision lists, which base their classifications on the single most confident feature and smooth feature ratios sensitive to context distance, are not similarly distracted by the competing attraction of weaker, often spurious, features from wide context. But they do not benefit from greater context either, with performance growth nearly flat with increasing context width.

Further insight may be gained by examining context-window sensitivity based on the part-of-speech of the keyword. As also observed by Yarowsky (1993), the marginally useful context width for nouns is quite large, plateauing here at ± 150

words for the Bayesian model family. Likewise, the marginally useful context width for adjectives is quite narrow, showing negligible contribution beyond the initial ± 5 context window. Interestingly, however, for verbs the marginally useful context width plateaus at an unexpectedly wide ± 60–80 words. This would suggest that the verbs in the English SENSEVAL2 inventory exhibit a substantial mass of sense ambiguities sensitive to broad topic as well as argument-based selectional preferences, which one would expect to have limited occurrences beyond a -5 to $+10$ word context window. This observation is further supported by the continued increase in performance when widening context beyond ± 25 words for decision lists, which base classifications on only the single most reliable contextual feature. Examples of individual SENSEVAL2 verbs which benefit most from the use of wide topic context include *to collaborate* (e.g. traitorous vs. ordinary cooperation) and *to train* (e.g. exercise, instruct, etc.).

3.4 Performance sensitivity to size of training data

Learning curves relative to size of training data are a longstanding foundation of machine learning evaluations. Figure 5 shows the performance obtained by each classifier on randomly selected variable-size subsets of the training data, including a minimum of one example per sense.[4] From the six classifiers, FENBayes exhibits the largest relative drop in performance on small training sizes. The baseline performance using the most likely sense is not flat here because the choice of the most likely sense was made on training data rather than test data. As training sizes increase, the accuracy of correctly identifying the most likely sense also increases. Cosine and Naïve Bayes still underperform this conservative lower-bound at small data sizes, while the discriminative models (TBL and DL) perform relatively well on small training data.

When replotting Figure 5(a) on a log-log scale (using error rate rather than accuracy) in Figure 5(b), a clear log-linear reduction in error rate emerges across an eight-fold increase in training data size. This shows that the performance difference between the best and worst-performing algorithm is comparable to that achieved by an approximately three-fold increase in training data at any point. While the relative cost-efficiency of additional training annotation versus comparable effort in algorithm development are open to debate and highly sensitive to one's relative cost models (Ngai and Yarowsky 2000), these curves at least show the potential for brute-force reduction of error rate through additional annotation. Unfortunately, assuming continued log-linear extrapolation, one would expect to require a 100-fold increase in training data for a brute-force performance increase to 80% accuracy (for the English lexical choice task), and a 5000-fold increase in training data for a brute-force increase to 90% mean accuracy. However, such long-range extrapolations can

[4] Note that the largest measured size here is 80% of the original training set, given that 20% of data was set aside as devtest in five-fold cross validation, leaving the final SENSEVAL2 test data untouched in these experiments to avoid the possibility of even indirect optimization of parameters and methods on this easy-to-overuse resource.

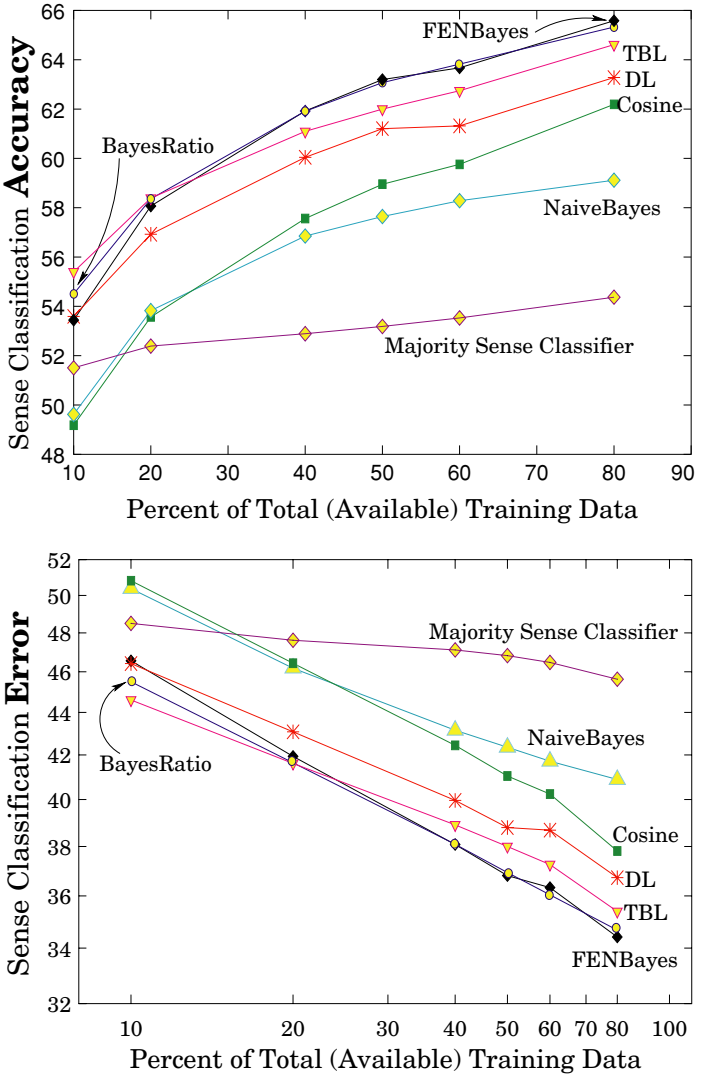


Fig. 5. Performance based on training data size.

be highly unreliable, and even a small log-slope change can substantially affect the anticipated performance increase from additional data.

3.5 Performance sensitivity to task difficulty

Algorithm performance also shows sensitivity to various measures of task difficulty, such as sense entropy and number of senses per keyword. Rather than alter the data artificially to create the desired change in target parameters, which might distort other data characteristics in unpredictable ways, Figures 6(a)–8(a) are based on the unaltered data set sorted into 15 equal-token-size bins by the focus parameter (with an average of five polysemous words and 573 tokens in each bin).

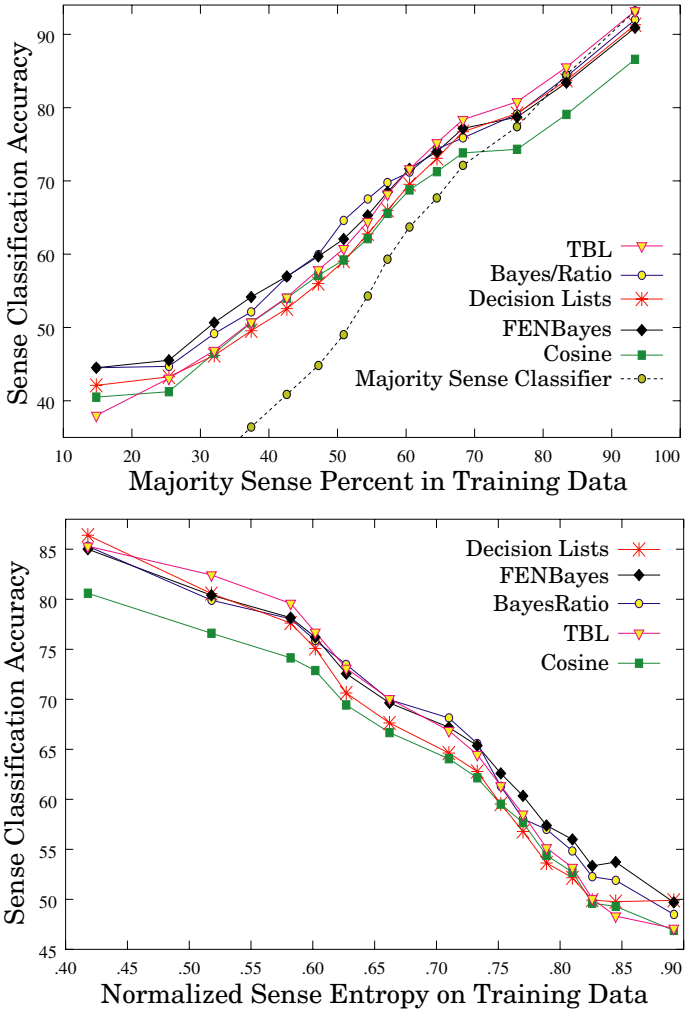


Fig. 6. Performance based on the probability of the majority sense and sense entropy.

Probability of the majority sense: Figure 6(a) shows the relative performance of algorithms given the probability of the majority sense in the training data. When the probability of the majority sense exceeds 80%, it appears difficult for any tested algorithm to exceed the performance of a baseline algorithm which always assigns the majority sense observed in the training data. In fact, cosine substantially underperforms this baseline in this range, suggesting that it insufficiently weights the sense prior. The discriminative models tend to do relatively well when the majority sense percentage is high (TBL significantly outperforms FENBayes on data where the majority sense exceeds 80% ($p \leq 0.05$), and does significantly worse than FENBayes when the majority sense is less than 40% ($p \leq 0.01$)).

Sense entropy: Figure 6(b) measures algorithm performance given the entropy of a keyword's sense distribution as a fraction of the total possible entropy

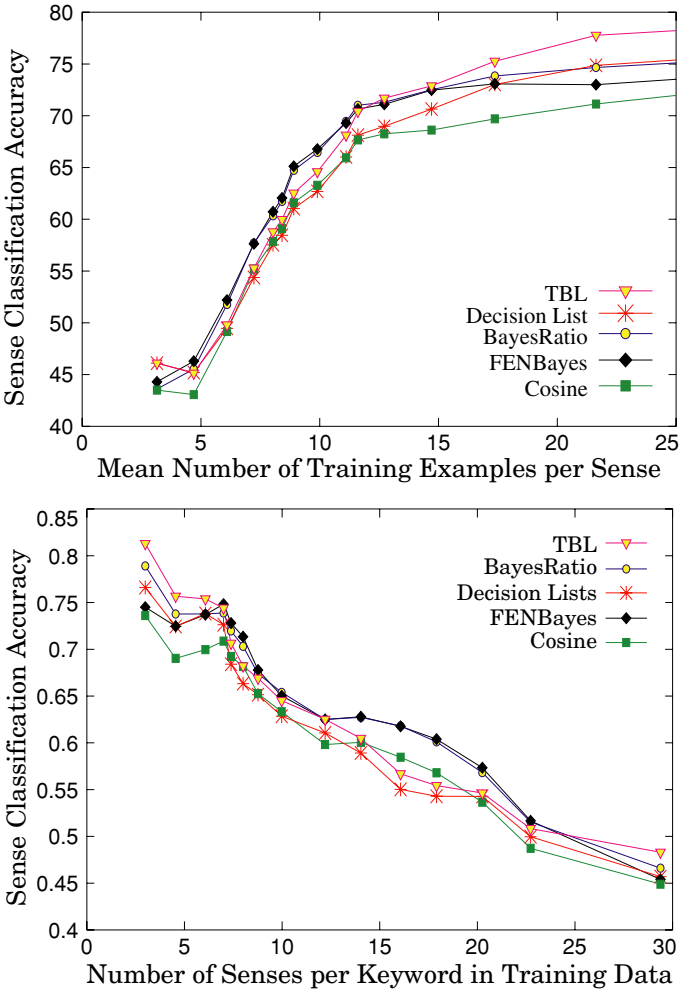


Fig. 7. Performance based on number of examples per sense and senses per keyword.

($H_r(p) = \frac{H(p)}{log_2(\#senses)}$.[5] The performance of all algorithms degrades substantially with higher sense entropy, although the discriminative models perform relatively well on lower entropy data (TBL significantly outperforms FENBayes when relative entropy <0.60 ($p \leq 0.05$), and significantly underperforms it when relative entropy >0.75 ($p \leq 0.001$)).

Mean number of training examples per sense: Figure 7(a) shows that all algorithms perform substantially better on data sets with with a greater mean number of training examples per sense. The discriminative models do relatively well when this number is high, with TBL significantly outperforming all aggregative models

[5] For example, an 8-sense keyword with an entropy of 1.5 would be divided by its maximum possible entropy ($log_2(8) = 3$), for a ratio of 0.5. This helps to focus the measure on high ambiguity of distribution over the possible senses, in contrast to simple entropy which largely tends to mirror the number of possible senses.

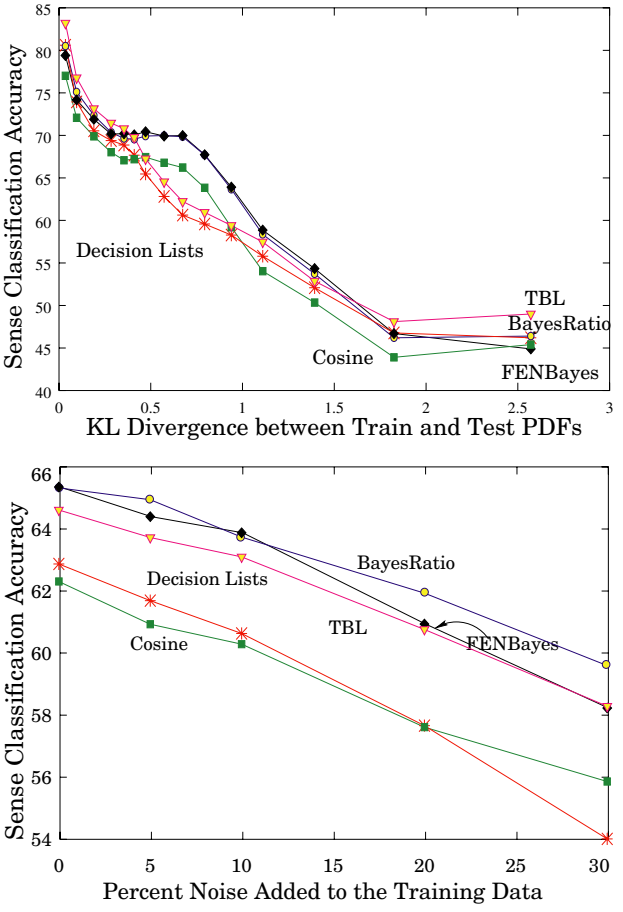


Fig. 8. Performance based on train/test divergence and noise in the training data.

when the number of training examples per sense is greater than 15 ($p \leq 0.01$), and significantly underperforming the Bayesian models when the mean number of training examples per sense is less than 12 ($p \leq 0.01$).

Number of senses: 7(b) shows algorithm performance on keyword data sorted by their number of senses. Consistent with trends illustrated in previous figures, TBL significantly outperforms the best aggregative algorithms (FENBayes and BayesRatio) on keywords with fewer than 8 senses ($p \leq 0.05$) and significantly underperforms these same algorithms when a keyword's number of senses is greater than 15 ($p \leq 0.05$).

Training/test divergence: Figure 8(a) shows that the performance of all algorithms degrades substantially when the divergence between the sense probability distribution in training and test data is high, as measured by Kullback-Leibler (KL) divergence. Performance degrades when senses expected to be rare based on training data are more common in test data, and vice-versa. This suggests a high reliance on prior probabilities in selecting senses, especially in ambiguous contexts. The discriminative models tend to do relatively well in cases of large training-test divergence; TBL

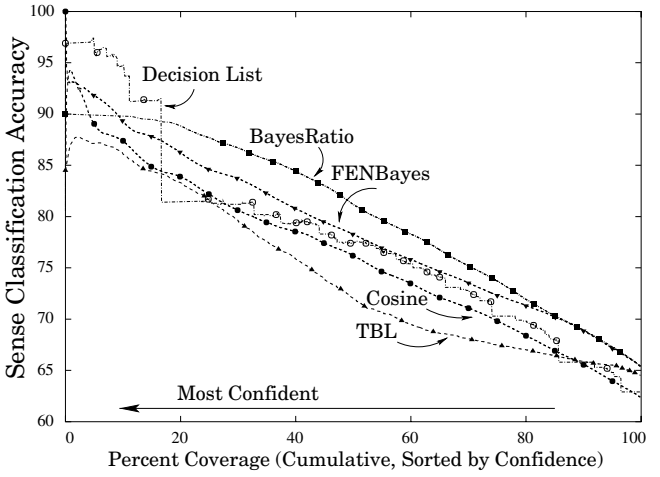


Fig. 9. Accuracy versus algorithm confidence.

significantly outperforms the best aggregative model (FENBayes) when divergence exceeds 1.5 ($p \leq 0.05$), and significantly underperforms it when divergence is less than 1.0 ($p \leq 0.05$).

Noise in training data: Figure 8(b) shows the relative tolerance of classification algorithms to training data noise, in terms of the percentage of randomly introduced changes to the training sense classifications (the altered senses are generated with a probability following the original sense distribution). Decision lists are particularly sensitive to training noise, given their dependence on single high-confidence features, which are weighted by their training data sense purity.

3.6 Performance sensitivity to sample coverage and system confidence

For many applications it may be preferable to achieve more accurate partial results at the expense of reduced coverage. Figure 9 shows the accuracy of each algorithm as a function of data coverage, when their classifications are considered in decreasing order of algorithm confidence (which was measured as the probability of the most likely sense on a test instance under the specified model).

One striking outcome is that Decision Lists are best able to identify their most accurate output. In the roughly 20% of test data where decision lists are internally most confident, their accuracy ranged from 92–97% and achieved a 40% lower error rate relative to other algorithms' most confident 20% of their output. This ability to more successfully identify their most accurate output makes decision lists particularly useful for techniques in unsupervised and minimally supervised learning (such as co-training) where the ability to anchor a bootstrapping procedure with high-confidence initial seed sets is useful (Yarowsky 1995). This property is also helpful in variable-weight classifier combination based on confidence. Accurately estimating the test samples where an algorithm is likely correct (or not) is important to successfully boosting or downweighting an algorithm's vote relative to an undifferentiated baseline voting weight.

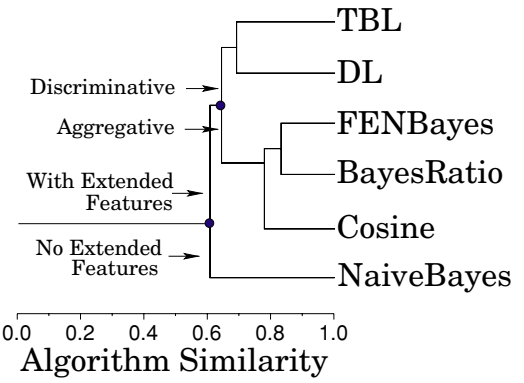


Fig. 10. Induced algorithm clustering.

TBL notably underperforms here, but this is a property of its pylon-like decision topology not being conducive to differential probability estimation. Traditional TBL (Brill 1995) was limited to a single probability output per output label. The TBL model used here incorporates Florian, Henderson and Ngai's (2000) algorithm for history-based probability estimation, which dramatically improves upon the TBL baseline but is still hindered by the constraints of the pylon classification topology, which merges diverse predicate histories to minimize training data fragmentation.

3.7 Analysis of patterns in algorithm performance

Throughout this paper, empirical differences in behavior have been noted between the class of *aggregative* algorithms (FENBayes, BayesRatio, Cosine) and *discriminative* algorithms (DL and TBL). This section will provide additional empirical and functional motivations for this algorithm classification typology.

Figure 10 shows a dendogram automatically generated by maximal-linkage hierarchical agglomerative clustering of these algorithms, where the similarity between two classifiers is measured by their pairwise agreement rate on the English SENSEVAL2 data. Interestingly, the observed top-level split is between those five algorithms which share the enhanced feature space (sensitive to relative position and syntactic relationship), and the traditional Naïve Bayes algorithm using only bag-of-words features. This indicates that major differences in the utilized feature space are more dominant factors in algorithm performance than differences in algorithm architecture. As illustrated in Figure 1, the 5–12% absolute difference in performance between Naïve Bayes and FENBayes (which differ only in their utilized feature space) significantly exceeds the mean cross-algorithm performance differences of all algorithm pairs on all four languages when using the same feature space ($p \leq 0.01$, with the exception of cosine on Basque). Furthermore, as illustrated in Figure 2, the cost of omitting any major feature type is a significant performance loss for all algorithms ($p \leq 0.05$), often exceeding cross-algorithm performance differences. These observations imply that the highest priority in algorithm design should be to enrich and improve the supporting feature space as much as possible.

However, when the available feature space is held constant, the discriminative algorithms (DL and TBL) and the aggregative algorithms (FENBayes, BayesRatio, Cosine) form natural subclusters in Figure 10 as hypothesized.

A brief comparison of functional algorithm behavior motivates this clustering. One of the most salient differences between the discriminative and aggregative algorithms is their exploitation of the feature space. As previously noted, decision lists base their classification on only the single most confident feature present in a test context. TBL also typically bases its classification on a relatively small set of the most incrementally informative contextual features in greedy, error-minimizing training. In contrast, the methods in the aggregative cluster (FENBayes, BayesRatio and Cosine) integrate all the observed contextual features into one consensus score. Thus the aggregative models should be more effective on contexts where several weak clues contribute towards a more confident aggregate classification. In addition, one would expect that the discriminative models should be relatively more capable in contexts where a single feature (in the case of DL) or limited combination of features (for TBL) is decisive.

The findings in sections 3.1–3.3 support these analyses. The discriminative models have significantly weaker performance on nouns relative to other classifiers, but perform competitively on adjectives and verbs (where very limited syntactic relationships are typically decisive). They are significantly less impaired by the restriction to local or syntactic features. They also perform significantly better when context width is restricted. Furthermore, section 3.5 shows that the discriminative models perform significantly better on lower entropy, fewer-sense-per-keyword data, and on data where the prior probability of the majority sense is high, while the aggregative models appear to better tolerate sense inventories with multiple, highly similar senses. This would indicate that discriminative models may benefit from use in a hierarchical, recursive splitting of major senses followed by subsenses, rather than a single flat k-way disambiguation of the full subsense inventory at once.

4 Implications and conclusion

A general conclusion one can draw from all of these previous results and observations is that there is no one-size-fits-all algorithm that excels at each of the diverse challenges in sense disambiguation. This is illustrated most clearly in Table 1 (detailing performance on each English keyword), where even weakly performing algorithms such as cosine and most frequent sense classifiers are top performers for several keywords. There is a remarkable diversity of success across algorithms. Comparable tables for our other target languages show similar diversity of effectiveness. Moreover, section 3 shows that the (empirically motivated) discriminative and aggregative algorithm classes often have complementary regions of effectiveness across numerous parameters. These results strongly motivate the usage of classifier combination algorithms to incorporate the diverse and unique strengths of these algorithms into a synergistic consensus. Such a goal is realized in the companion article to this study (Florian *et al*., this issue), which investigates and comprehensively evaluates a range of traditional and novel classifier combination

algorithms. The conclusion of that study is that robust combination of these diverse classifiers can achieve significant improvement over the single-best stand-alone classifier, achieving the highest known current performance on the SENSEVAL2 lexical sample tasks for English, Spanish, Swedish and Basque.

In complementary contrast, the work presented above has yielded insight into the nature, scope and implications of the diversity found in these component algorithms and their underlying phenomena. This includes the observation that the quality of the feature space can have significantly greater impact on WSD performance than the choice of classification algorithm. Collectively, it constitutes the most comprehensive survey of evaluation measures, languages, algorithms and diverse parameter spaces yet applied to word sense disambiguation in a single unified experimental framework.

References

Brill, E. (1995) Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics* **21**(4): 543–565.

Cucerzan, S. and Yarowsky, D. (2002) Augmented mixture models for lexical disambiguation. *Proceedings EMNLP'02*, pp. 33–40.

Edmonds, P. and Cotton, S. (2001) SENSEVAL-2: Overview. *Proceedings* SENSEVAL-2 , pp. 1–6.

Florian, R. and Yarowsky, D. (1999) Dynamic nonlocal language modeling via hierarchical topic-based adaptation. *Proceedings ACL99*, pp. 167–174.

Florian, R., Henderson, J. C. and Ngai, G. (2000) Coaxing confidence from an old friend: Probabilistic classifications from transformation rule lists. *Proceedings EMNLP 2000*, pp. 26–34. Hong Kong.

Florian, R., Cucerzan, S. P., Schafer, C. and Yarowsky, D. (2002) Classifier combination for word sense disambiguation. *J. Natural Lang. Eng.* **8**(4).

Gale, W., Church, K. and Yarowsky, D. (1992) A method for disambiguating word senses in a large corpus. *Comput. and the Humanities* **26**: 415–439.

Kilgarriff, A. and Palmer, M. (2000) Introduction to the special issue on senseval. *Comput. and the Humanities* **34**(1): 1–13.

Kilgarriff, A. and Rosenzweig, J. (2000) Framework and results for English Senseval. *Comput. and the Humanities* **34**(1): 15–48.

Leacock, C., Towell, G. and Voorhees, E. (1993) Corpus-based statistical sense resolution. *Proceedings ARPA'93*, pp. 260–265.

Mooney, R. (1996) Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. *Proceedings EMNLP'96*, pp. 82–91.

Ng, H. T. (1997) Getting serious about word sense disambiguation. *Proceedings SIGLEX97*, pp. 1–7.

Ngai, G. and Yarowsky, D. (2000) Rule writing or annotation: cost-efficient resource usage for base noun phrase chunking. *Proceedings ACL'02*, pp. 117–125.

Pedersen, T. (2001) A decision tree of bigram s is an accurate predictor of word sense. *Proceedings NAACL'01*, pp. 79–86.

Stevenson, M. and Wilks, Y. (2001) The interaction of knowledge sources in word sense disambiguation. *Computational Linguistics* **27**(3): 321–349.

Yarowsky, D. (1993) One sense per collocation. *Proceedings, ARPA Human Language Technology Workshop*, pp. 266–271. Princeton, NJ.

Yarowsky, D. (1995) Unsupervised word sense disambiguation rivaling supervised methods. *Proceedings ACL-95*, pp. 189–196.

Yarowsky, D. (1996) Homograph disambiguation in speech synthesis. In: Olive, J., van Santen, J., Sproat, R. and Hirschberg, J., editors, *Progress in Speech Synthesis*, pp. 159–175. Springer-Verlag.