Hierarchical Decision Lists for Word Sense Disambiguation

David Yarowsky
Dept. of Computer Science
and Center for Language and Speech Processing
Johns Hopkins University
Baltimore, MD 21218
yarowsky@cs.jhu.edu

Abstract. This paper describes a supervised algorithm for word sense disambiguation based on hierarchies of decision lists. This algorithm supports a useful degree of conditional branching while minimizing the training data fragmentation typical of decision trees. Classifications are based on a rich set of collocational, morphological and syntactic contextual features, extracted automatically from training data and weighted sensitive to the nature of the feature and feature class. The algorithm is evaluated comprehensively in the SENSEVAL framework, achieving the top performance of all participating supervised systems on the 36 test words where training data is available.

Keywords: word sense disambiguation, decision lists, supervised machine learning, lexical ambiguity resolution, SENSEVAL

1. Introduction

Decision lists have been shown to be effective at a wide variety of lexical ambiguity resolution tasks including word sense disambiguation (Yarowsky, 1994, 1995; Mooney, 1996; Wilks and Stevenson, 1998), text-to-speech synthesis (Yarowsky, 1997), multilingual accent/diacritic restoration (Yarowsky, 1994), named entity classification (Collins and Singer, 1999) and spelling correction (Golding, 1995).

One advantage offered by interpolated decision lists (Yarowsky, 1994, 1997) is that they avoid the training data fragmentation problems observed with decision trees or traditional non-interpolated decision lists (Rivest, 1987). They also tend to be effective at modelling a large number of highly non-independent features that can be problematic to model fully in Bayesian topologies for sense disambiguation (Gale, Church and Yarowsky, 1992; Bruce and Wiebe, 1994).

This paper presents a new learning topology for sense disambiguation based on *hierarchical decision lists*, adding a useful degree of conditional branching to the decision list framework. The paper also includes a comprehensive evaluation of this algorithm's performance on extensive previously unseen test data in the SENSEVAL framework (Kilgarriff, 1998; Kilgarriff and Palmer, 1999), showing its very successful application to the complex and fine-grained HECTOR sense inventory.

2. System Description

The basic decision-list algorithms used in this system are described in Yarowsky (1994, 1997), with key details outlined below. Note that part-of-speech (POS) tagging is treated as a disjoint task from sense tagging, and a trigram POS tagger has been applied to the data first. The POS tagger has not been optimized for the specific idiosyncrasies of the SENSEVAL words and such optimization would likely be helpful.

2.1. Feature Space

The contextual clues driving the decision list algorithm are a cross-product of rich sets of token types and positions relative to the keyword. The example decision lists in Table I illustrate a partial set of such features. Positional options include relative offsets from the keyword (+1,−1,−2), the keyword itself (+0), co-occurrence within a variable k-word window ($\pm k$), and larger n-gram patterns (+1+2, -1+1). Another crucial positional class are the wide range of syntactic relations extracted from the data using an island-centered finite state parser. The valid patterns differ depending on keyword part of speech, and for nouns they are V/OBJ - the verb of which the keyword is an object (e.g. *showed very abundant promise*), SUBJ/V - the verb of which the keyword is the subject, and MODNOUN - the optional headnoun modified by the noun. Each of these patterns help capture and generalize sets of very predictive longer-distance word associations.

Five major token types are measured in each of the diversity of syntactic/collocational positions, including W=literal word, L=lemma (*win/V=win,wins,won,winning*), P=part-of-speech, C=word class (e.g. *countryname*) and Q=question, such as is the word in the given position capitalized? Together this rich cross-product of word-type and syntactic position offers considerable refinement over the bag-of-words model.

2.2. Feature Weighting and Basic Decision List Generation

For each word+position feature f_i, a smoothed log-likelihood ratio ($\frac{P(f_i|s_j)}{P(f_i|\neg s_j)}$) is computed for each sense s_j, with smoothing based on an empirically estimated function of feature type and relative frequency. Candidate features are ordered by this smoothed ratio (putting the best evidence first), and the remaining probabilities are computed via the interpolation of the global and history-conditional probabilities.[1]

[1] The history-conditional probabilities are based on the residual data for which no earlier pattern in the decision list matches. While clearly more relevant, they are often much more poorly estimated because the size of the residual training data

2.3. Hierarchical Decision Lists

One limitation of traditional flat decision lists is that they do not support conditional branching. Yet it is often the case that given some major splitting criterion (such as whether a keyword is identified as a noun or verb) we would wish to divide the control flow of the decision procedure into relatively independent paths specialized for the modelling needs of each side of the splitting partition. Decision trees, which entail complete path independence after *every* node split, pay for this power with wasteful training data fragmentation. Yet a simple forest of uniflow decision lists fails to capture the common hierarchical structure to many decision problems. This proposed hybrid supports several major useful decision flow partitions, but largely retains the uniflow non-data-fragmenting benefits of interpolated decision lists.

The key to the success of this approach is defining a class of such natural major partitioning questions for the application, and pursuing exhaustive cross-validated search on whether any candidate partition improves modelling of the training data.[2]

For the application of sense disambiguation, some natural major decision-flow partitioning criteria are:

- *Split on the part of speech of the keyword.* As previously noted, the sense inventory and natural decision lists for the noun and verb senses of words is widely divergent, and thus a top-level split in control flow based on keyword part-of-speech is very natural. The top-level decision list in Table I illustrates this split into subsequent LN (noun) and LV (verb) decision lists for the word *promise*.

- *Split on keyword inflection.* Similarly, within a major part-of-speech, different inflectional forms (e.g. *promise* and *promises*, or *scrap* and *scraps*) often exhibit different sense inventory distributions and different optimal subsequent modeling. In the midlevel list in Table I, *promises*(NOUN) separately yields a mostly pure sense distribution that effectively excludes senses 5 and 6. In contrast,

shrinks at each line of the decision list. A reasonable compromise is to interpolate between two conditional probabilities for any given feature f_i at line i of the list, $\beta_i P(s_j|f_i) + (1-\beta_i)P(s_j|\neg f_1 \wedge .. \wedge \neg f_{i-1})$, where β_i is optimized on training data, optionally sensitive to relative sample sizes. The case where $\beta_i = 0$ corresponds to the original Rivest (1987) decision list formulation.

[2] Training time for a single linear decision list is typically under 2 seconds total elapsed clock time on a SPARC Ultra-2. Because there is often a natural hierarchical sequence of split question types, and because many combinations are unnecessary to consider (e.g. *nmod* and noun inflectional cases under the top-level LV=verb split), the total space of tested split combinations is typically (much) less than 1000, and hence very computationally tractable.

the singular inflection *promise*(NOUN) retains this ambiguity, requiring the subsequent decision list L4 to distinguish senses 4, 5 and 6. While this partition could technically have been done with finer grained parts of speech at the top-level split, the interaction with other mid-level questions (see below) makes this two-tiered part-of-speech partition process worthwhile.

- *Split on major idiomatic collocations.* Many idiomatic collocations like *keep/break/give/make a promise* or *shake up/down/out/off* benefit from a subsequent specialized decision list to resolve the possible sense differences for this specific collocation (e.g. L1 or L2), and when corresponding to a single sense number (e.g. *keep a promise* $\rightarrow$ 4.3) can directly yield a sense-tag output (as a specialized decision list would have no residual ambiguity to resolve). Such candidate collocations are extracted from the basic defining inventory `mne-uid.map`[3] (e.g. *promise 538409 keep n promise//4.3*) and/or from collocations that are found to be tightly correlated with specialized sense numbers in the training data. The decision to split out any such collocation is based on an empirical test of the global efficacy of doing so.[4]

- *Split on syntactic features.* In many cases it is also useful to allow mid-level splits on syntactic questions such as whether a keyword noun premodifies another noun (e.g. the standard syntactic feature MODNOUN != NULL). Such a split is not useful to *promise*, but is widely applicable to the HECTOR inventory given its tendency to make an NMOD subsense distinction.

- *Partition subsenses hierarchically.* When a sense inventory has a deep sense/subsense structure, it may be useful to have third-level decision lists focus on major sense partitions (e.g. 4/5/6) and when appropriate yield pointers to a finer-grained subsense-resolving decision list (e.g. L5 = 5.1/5.2/5.3). This multi-level subsense resolution is most effective when the subsenses are tightly related to each other and quite different from the other major senses. For performance reasons, however, a flat direct subsense

[3] `http://www.itri.bton.ac.uk/events/senseval/mne-uid.map`

[4] Note that small numbers of the *make/give/break a promise* senses 4.1, 4.2 and 4.3 are not caught by the specialized patterns in the mid-level decision list. There are several reasons for this. A majority of these few misses are due to parsing errors that failed to recognize the correct headword given unusually convoluted syntax. In some cases, there may be genuine ambiguity, as in sentence 800848 "that the *promises* given to him be kept", which is recognized as *4.2 = give a promise* but was human labelled as *4.3 = keep a promise.*

Table I. Partial decision list hierarchy for the SENSEVAL word **promise**

Top-level Decision List for **promise**

Pattern Loc	Typ	Token		Next List	1	3	4	4.1	4.2	4.3	4.4	5	6
+0	P	NOUN	→	*LN*(⇓)	0	0	297	53	5	37	11	22	93
+0	P	VERB	→	LV	440	115	0	0	0	0	0	0	0

(Columns 1–6: Empirical Sense Distribution)

⇓

Mid-level Decision List for **promise.LN** (noun)

Pattern Loc	Typ	Token		Next List	4	4.1	4.2	4.3	4.4	5	6
V/obj	L	keep/V	→	**4.3**	0	0	0	31	0	0	0
V/obj	L	break/V	→	**4.4**	0	0	0	0	11	0	0
V/obj	L	make/V	→	L1	2	44	0	0	0	0	2
V/obj	L	give/V	→	L2	0	0	5	1	0	1	2
+0	W	promises	→	L3	115	5	0	0	0	0	1
+0	W	promise	→	*L4*(⇓)	180	3	0	1	0	21	88

(Columns 4–6: Empirical Sense Distribution)

⇓

(Abbreviated) Terminal Decision List for **promise.L4** (promise-noun-singular)

Pattern Loc	Typ	Token		Output Sense	LogL	4	4.1	4.2	4.3	4.4	5	6
+1	W	to	→	**4**	9.51	41	0	0	0	0	0	0
-1	W	of	→	**6**	8.16	0	0	0	0	0	0	12
-1	L	early/J	→	**6**	7.38	0	0	0	0	0	0	7
V/obj	L	show/V	→	**6**	7.27	0	0	0	0	0	0	13
+1	W	at	→	**6**	6.16	0	0	0	0	0	0	3
-1	L	firm/J	→	**4**	5.74	6	0	0	0	0	0	0
+1	L	do/V	→	**4**	5.70	3	0	0	0	0	0	0
-1	W	such	→	**6**	5.57	0	0	0	0	0	0	2
-1	W	much	→	**6**	5.57	0	0	0	0	0	0	2
+1	W	when	→	**6**	5.57	0	0	0	0	0	0	2
+1	W	on	→	**6**	5.57	0	0	0	0	0	0	2
+1	W	as	→	**6**	5.57	0	0	0	0	0	0	2
-1	W	your	→	**4**	5.16	2	0	0	0	0	0	0
+1	W	during	→	**4**	5.16	2	0	0	0	0	0	0
$\pm k$	L	free/J	→	**4**	4.74	15	0	0	0	0	0	0
V/obj	L	trust/V	→	**4**	4.74	3	0	0	0	0	0	0
$\pm k$	L	support/N	→	**4**	4.64	14	0	0	0	0	0	0
$\pm k$	L	election/N	→	**4**	4.29	11	0	0	0	0	0	0
subj/V	L	contain/V	→	**4**	4.18	2	0	0	0	0	0	0
V/obj	L	win/V	→	**4**	4.16	2	0	0	0	0	0	0
V/obj	L	repeat/V	→	**4**	4.16	2	0	0	0	0	0	0
V/obj	L	honour/V	→	**4**	4.16	2	0	0	0	0	0	0
-1	L	rhetorical/J	→	**5**	4.09	0	0	0	0	0	1	0
-1	L	increase/V	→	**5**	4.09	0	0	0	0	0	1	0
-1	L	future/J	→	**5**	4.09	0	0	0	0	0	1	0

(Columns 4–6: Empirical Sense Distribution) partition (5.1/5.2/5.3/6.1/6.2) was generally pursued on the SENSEVAL data. Recent results indicate that an even more effective compromise in this case is to utilize a deeply hierarchical approach where probabilities are interpolated across sibling subtrees.

3. Evaluation and Conclusion

Table II details the performance of the JHU hierarchical decision list system in the 1998 SENSEVAL evaluation. To help put the performance figures in perspective, the average precision for all supervised systems is given, as is the precision for the best performing system of any type. All data to the left of the vertical line are based on the July 98 bakeoff. Here the JHU system achieved the highest average overall precision on the 36 "trainable" words (for which tagged training data was available).

Due to the haste under which the primary evaluation was conducted, and the inability to manually check the output, for three words (*bet*/Noun, *floating*/Adj and *seize*/Verb) the JHU system had errors in mapping from its internal sense number representations (a contiguous sequence 0,1,2,3,...) to the standard output sense IDs (538411, 537573, 537626, etc.). This resulted in significantly lower scores for these three words. Thus for the 2nd round October 98 evaluations, these simple mapping errors were corrected and nothing else was changed. Corrected performance figures are given to the right of the vertical line.

The additional evaluation area consisted of the 5 words for which no annotated training data was available. As a demonstration of robustness, the JHU *supervised* tagger was applied to these words as well, trained only on their dictionary definitions. Precision for these words was measured at deaf=94.3, disability=90.0, hurdle=69.0, rabbit=76.5 and steering=95.6, with an overall average precision of 81.7%, the 2nd-highest untrainable-word score among all participants, including those systems specialized for unsupervised and dictionary-based training.

Finally, the comparative advantage of hierarchical decision lists relative to flat lists was investigated. Using the most fine-grained inventory scoring and 5-fold cross validation on the training corpus for these additional studies, average accuracy on the 36 test words dropped by 7.3% when the full 3-level lists were replaced by a single 2-level list splitting only on the part of speech of the keyword. A further 1% drop in drop in average accuracy was observed on the 'p' words (*bitter*, *sanction*, etc.) when their top-level POS split was merged as well.[5] Taken together

[5] One explanation for this smaller drop is that the feature spaces for different parts of speech are somewhat orthogonal, making it relatively less costly to accommodate their separate decision threads in the same list.

Table II. Performance of the JHU system on the 36 trainable words

Word	POS	Avg. Syst Prec.	Initial JHU Prec.	Best Syst Prec.	JHU Rank of 21	JHU % of Best	Final JHU Prec.	New JHU Rank
All Trainable	a	72.7	**77.8**	77.8	1	100.0	**77.3**	1
All Trainable	n	81.7	**84.7**	87.0	3	97.4	**87.0**	2
All Trainable	p	73.7	**78.1**	78.1	1	100.0	**78.1**	1
All Trainable	v	66.4	**73.4**	73.4	1	100.0	**74.3**	1
All Trainable	**all**	73.4	**78.4**	78.4	1	100.0	**78.9**	1
accident	n	92.3	**95.6**	95.7	2	99.9		
amaze	v	94.6	**100.0**	100.0	1	100.0		
band	p	87.5	**90.6**	90.6	1	100.0		
behaviour	n	95.8	**96.1**	96.4	+	99.7		
bet	n	60.8	**52.2**	75.7	-	69.0	**78.8**	1
bet	v	55.5	**69.8**	78.6	3	88.8		
bitter	p	63.8	**64.9**	73.4	+	88.4		
bother	v	75.3	**80.2**	86.5	2	92.7		
brilliant	a	56.1	**59.5**	61.4	3	96.9		
bury	v	47.8	**46.2**	57.3	+	80.6		
calculate	v	87.9	**92.2**	92.2	1	100.0		
consume	v	52.1	**53.0**	58.5	+	90.6		
derive	v	59.5	**66.4**	67.1	2	99.0		
excess	n	83.5	**87.8**	90.0	2	97.6		
float	n	65.1	**82.2**	82.2	1	100.0		
float	v	47.1	**54.0**	61.4	2	87.9		
floating	a	57.2	**0.0**	80.9	-	0.0	**63.6**	+
generous	a	53.7	**59.5**	61.2	2	97.2		
giant	a	84.1	**99.1**	99.5	3	99.6		
giant	n	83.6	**85.8**	91.0	+	94.3		
invade	v	56.5	**54.6**	63.4	-	86.1		
knee	n	81.5	**84.6**	87.1	2	97.1		
modest	a	68.2	**71.8**	72.9	+	98.5		
onion	n	86.7	**92.1**	92.5	2	99.6		
promise	n	84.4	**88.6**	88.6	1	100.0		
promise	v	69.8	**90.9**	91.3	2	99.6		
sack	n	76.9	**87.8**	87.8	1	100.0		
sack	v	83.1	**97.8**	97.8	1	100.0		
sanction	p	76.9	**86.5**	86.5	1	100.0		
scrap	n	64.2	**75.1**	79.5	2	94.5		
scrap	v	78.7	**94.9**	95.1	2	99.8		
seize	v	64.2	**65.3**	68.4	2	95.5	**73.5**	1
shake	p	68.6	**70.9**	76.5	3	92.7		
shirt	n	90.8	**92.6**	97.8	3	94.7		
slight	a	92.0	**96.3**	96.3	1	100.0		
wooden	a	95.8	**97.4**	98.0	3	99.4		

Rank/best for all systems — + = above median rank
Average precision for supervised systems — - = below median rank

these results indicate that optionally splitting dataflow on keyword inflections, major syntactic features, idiomatic collocations and subsenses and treating these in separate data partitions can improve performance while retaining the general dataflow benefits of decision lists.

One natural next step in this research is to evaluate the minimally supervised bootstrapping algorithm from Yarowsky (1995) on this data. Results on the word *rabbit* show a 24% increase in performance using bootstrapping on unannotated *rabbit* data over the supervised baseline. The major impediment to this work is the lack of discourse ID's in the data (or at least a matrix indicating those test sentences co-occurring in the same discourse). This information is crucial to the co-training of the one-sense-per-collocation and one-sense-per-discourse tendencies that enables the bootstrapping algorithm to gain new beachheads and robustly correct acquired errors or over-generalizations. Thus acquisition of some type of discourse or document IDs for the HECTOR sentences would potentially be a very rewarding investment.

References

Bruce, R. and J. Wiebe: 1994, Word-sense disambiguation using decomposable models. In *Proceedings of ACL '94*, pp. 139-146, Las Cruces, NM.

Collins, M. and Y. Singer: 1999, Unsupervised models for named entity classification. *Proc. of the 1999 Joint SIGDAT Conference*, pp. 100-110, College Park, MD.

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

Golding, A.: 1995, A Bayesian hybrid method for context-sensitive spelling correction. In *Proceedings of the 3rd Workshop on Very Large Corpora*, pp. 39–53.

Kilgarriff, A.: 1998, SENSEVAL: An exercise in evaluating word sense disambiguation programs. In *Proceedings of LREC*, pp. 581–588, Granada.

Kilgarriff, A. and M. Palmer (eds.): 2000, Special double issue on SENSEVAL. *Computers and the Humanities*, 33:4-5.

Mooney, R.: 1996, Comparative experiments on disambiguating word senses: An illustration of the role of bias in machine learning. In *Proc. of the Conference on Empirical Methods in Natural Language Processing*, pp. 82–91, Philadelphia.

Rivest, R: 1987, Learning decision lists. *Machine Learning*, 2:229–246.

Wilks, Y. and M. Stevenson: 1998, Word sense disambiguation using optimised combinations of knowledge sources. In *Proceedings of COLING/ACL-98*.

Yarowsky, D.: 1994, Decision lists for lexical ambiguity resolution: application to accent restoration in Spanish and French. *Proceedings of ACL '94*, pp. 88–95.

Yarowsky, D.: 1995, Unsupervised word sense disambiguation rivaling supervised methods. In *Proceedings of ACL '95*, pp. 189–196.

Yarowsky, D.: 1997, Homograph disambiguation in speech synthesis. In J. van Santen, R. Sproat, J. Olive and J. Hirschberg (eds.), *Progress in Speech Synthesis*, Springer-Verlag, pp. 159–175.