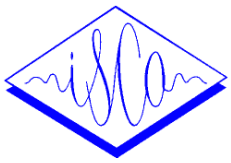

Second ESCA/IEEE Workshop
on Speech Synthesis
Mohonk, New Paltz, NY, USA
September 12-15, 1994

Homograph Disambiguation in Speech Synthesis

David Yarowsky
Department of Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104
yarowsky@unagi.cis.upenn.edu

Abstract

This paper presents a statistical decision procedure for lexical ambiguity resolution in speech synthesis. Based on decision lists, the algorithm incorporates both local syntactic patterns and more distant collocational evidence, combining the strengths of decision trees, N-gram taggers and Bayesian classifiers. The algorithm is applied to 7 major types of ambiguity where context is used to choose a word's pronunciation.

1 Problem Description

In speech synthesis, one frequently encounters words and numbers with multiple pronunciations that require the analysis of context for resolution. Seven major types of these homographs will be addressed here:

1) *Different Part of Speech*: Typically pronunciation ambiguities that result from differences in part of speech are resolved by a stochastic tagger such as [2]. However, in many cases such as *lives* (lɪvz/laɪvz) or *read* (ri:d/rɜd), performance can be improved considerably by modeling word-specific collocational tendencies.

2) *Same Part of Speech*: Words such as *bass* and *bow* exhibit different pronunciations with the same part of speech, and thus require additional "semantic" evidence for disambiguation.

3) *Proper Names* such as *Nice* and *Begin* are ambiguous in capitalized contexts, such as sentence initial position, titles and single-case text.

4) *Roman Numerals* are pronounced differently in contexts such as *Chapter III* and *Henry III*.

5) *Fractions/Dates* such as *5/16* may be pronounced as *five-sixteenths* or *May 16th*.

6) *Years/Quantifiers*: Numbers such as *1750* tend to be pronounced as *seventeen-fifty* when used as dates, and *one-thousand seven-hundred and fifty* when used before measure words such as *1750 miles*. Related cases include the distinction between *a 727 pilot* and *727 people*.

7) *Abbreviations* may exhibit multiple pronunciations, such as *St.* (Saint or Street) and *Dr.* (Doctor or Drive).

2 Algorithm

Previous approaches to lexical ambiguity resolution exhibit a number of weaknesses. Although N-gram taggers [2] model local syntactic sequences well, they are not adequate for modeling long-distance word-associations. Bayesian classifiers have been used for a number of sense disambiguation tasks [3] and capture wide-context lexical collocations well. However, they are ill-suited for modeling conditional dependencies such as word or class sequences. They require the assumption or independence or the complex modeling of statistical dependencies, problematic for non-independent feature sets using multiple levels of representation. Decision Trees [1] encounter severe difficulties with very large parameter spaces, such as the highly lexicalized feature sets frequently used in homograph resolution.

The algorithm described here is a hybrid approach, combining the strengths of each of these 3 methods. It was proposed in [5] and refined

in [8]. It is based on the formal model of Decision Lists in [4], although feature conjuncts have been restricted to a much narrower complexity, namely word and class trigrams. The algorithm models both local sequence and wide context well, and successfully exploits highly non-independent feature sets.

Lack of space limits the following description to an outline; see [8] for more detail. The algorithm will first be illustrated by the case of the individual homographs *lead* (lid/lɜd) and *bass* (beIs/bæs). The same procedure will then be applied to a large class of homographs such as fractions vs. dates.

The driving force behind this disambiguation algorithm is the uneven distribution of *collocations* (word associations) with respect to the ambiguous token being classified. For example, the following table indicates that word associations in various positions relative to the ambiguous token *bass* (including co-occurrence within a $\pm k$ word window) exhibit considerable discriminating power[1].

Position	Collocation	beIs	bæs
±k W	*fish* (in $\pm k$ words)	0	142
±k W	*guitar* (in $\pm k$ words)	136	0
+1 W	bass *player*	105	0
-1 W	*sea* bass	0	47

The goal of the initial stage of the algorithm is to measure a large number of collocational distributions and select those which are most useful in identifying the pronunciation of the ambiguous word. In addition to raw word associations, additional evidence considered includes collocations of lemmas (morphological roots), which tend to yield more succinct and generalizable evidence than inflected forms, and part-of-speech sequences, which capture syntactic rather than semantic distinctions in usage. A richer set of positional relationships beyond adjacency and co-occurrence in a window is also considered, including trigrams and (optionally) verb-object pairs. The following table indicates the pronunciation distributions observed for the noun *lead* for these various types of evidence:

Position	Collocation	lɜd	lid
+1 L	lead *level/N*	219	0
-1 W	*narrow* lead	0	70
+1 W	lead *in*	207	898
-1W,+1W	*of* lead *in*	162	0
-1W,+1W	*the* lead *in*	0	301
+1P,+2P	lead , <*NOUN*>	234	7
±k W	*zinc* (in $\pm k$ words)	235	0
±k W	*copper* (in $\pm k$ words)	130	0
-V L	*follow/V* + lead	0	527
-V L	*take/V* + lead	1	665

The discriminating strength of each piece of evidence is measured by the log-likelihood ratio:

$$Abs(Log(\frac{P(Pronunciation_1|Collocation_i)}{P(Pronunciation_2|Collocation_i)}))$$

Decision List for *lead (noun)*		
LogL	Evidence	Pron.
11.40	*follow/V* + lead	⇒ lid
11.20	*zinc* in $\pm k$ words	⇒ lɜd
11.10	lead *level/N*	⇒ lɜd
10.66	*of* lead *in*	⇒ lɜd
10.59	*the* lead *in*	⇒ lid
10.51	lead *role*	⇒ lid
10.35	*copper* in $\pm k$ words	⇒ lɜd
10.16	lead *poisoning*	⇒ lɜd
8.55	*big* lead	⇒ lid
8.49	*narrow* lead	⇒ lid
7.76	*take/V* + lead	⇒ lid
5.99	lead , *NOUN*	⇒ lɜd
1.15	lead *in*	⇒ lid

All measured collocations are sorted by this value, yielding a decision list with the strongest

[1] Such skewed distributions are in fact quite typical. A study in [7] showed that $P(pronunciation|collocation)$ is a very low entropy distribution. Certain types of content-word collocations seen only *once* in training data predicted the correct pronunciation in held-out test data with 92% accuracy.

and most reliable evidence listed first[2].

New examples are assigned pronunciations by using *only* the first line in the list that matches the target context. This differs significantly from the traditional procedure used in Bayesian classifiers of combining *all* matching evidence in a sum of log-likelihoods. There are several motivations for this simplifying approach. The first is that combining all available evidence rarely produces a different classification than just using the single most reliable evidence, and when these differ it is as likely to hurt as to help. A study in [8] based on 20 homographs showed that the two approaches agreed in 98% of the test cases. In the 2% cases of disagreement, using only the single best evidence performed slightly better than combining evidence. Of course this behavior does not hold for all classification tasks, but *does* seem to be characteristic of lexically-based word classifications. This may be explained by the empirical observation that in most cases, and with high probability, words exhibit only one sense in a given collocation[7].

Thus for this type of ambiguity resolution, there is no apparent detriment, and some apparent performance gain, from using only the single most reliable evidence in a classification. There are other advantages as well, including run-time efficiency and ease of parallelization. However, the greatest gain comes from the ability to incorporate multiple, non-independent information types in the decision procedure. A given word in context may match several times in the decision list, once for its parts of speech, lemma, inflected form, bigram, trigram, and possible word-classes as well. By only using one of these matches, the gross exaggeration of probability from combining all of these non-independent log-likelihoods is avoided. While these dependencies may be modeled and corrected for in Bayesian formalisms, it is difficult and costly to do so. Using only one log-likelihood ratio without combination frees the algorithm to include a wide spectrum of highly non-independent information without additional algorithmic complexity or performance loss.

Decision Lists for Ambiguity Classes: This algorithm may also be directly applied to large classes of ambiguity, such as distinguishing between fractions and dates. Rather than train individual pronunciation discriminators for *5/16* and *5/17*, etc., training contexts are pooled for all individual instances of the class. Since the disambiguating characteristics are quite similar for each class member, enhanced performance due to larger training sets tends to compensate for the loss of specialization[3]. A highly abbreviated decision list for the fraction/date class is shown below.

Decision List for Fraction/Date Class		
LogL	Evidence	Pronunciation
8.84	*<NUMBER>* (X/Y)	⇒ FRACTION
7.79	(X/Y) *<NOUNPL>*	⇒ FRACTION
7.58	(X/Y) *of*	⇒ FRACTION
6.79	*Monday* in $\pm k$ words	⇒ DATE
6.05	*Mon* in $\pm k$ words	⇒ DATE
5.96	(X/Y) *mile*	⇒ FRACTION
5.68	(X/Y) *inch*	⇒ FRACTION
4.22	*on* (X/Y)	⇒ DATE
3.96	*from* (X/Y) *to*	⇒ DATE

3 Evaluation

The following table provides a summary of the algorithm's performance on the classes of ambi-

[2]As most evidence includes zeros in this ratio, clearly some smoothing is necessary. Optionally, the probability estimates may be improved by interpolating between those computed from the full data set (the *global* probabilities) and those computed from the residual training data left at a given point in the decision list when all higher-ranked patterns failed to match (i.e. the *residual* probabilities). Finally, overmodeling, redundancy by subsumption, and redundancy by association may all be reduced by an additional pruning phase. See [8] for details of these procedures.

[3]Note that these lists are created assuming an equal prior probability. If space allows, highly skewed priors for individual class members such as *1/2* may be stored, and used during classification by appropriately adjusting the definition of highest ranking pattern.

guity studied. See [5] for a breakdown of performance on individual homographs in the first 3 cases. Data were extracted from a 400 million word corpus, including news stories, parliamentary debates and scientific text, augmented with examples from USENET news postings and e-mail correspondence[4]. Evaluation in each case is based on 5-fold cross-validation using held-out test data for a more accurate estimate of system performance.

System Performance		
Type of Ambiguity (Examp.)	Prior Prob.	% Corr.
Diff. Part of Speech (lives)	62	98
Same Part of Speech (bass)	72	97
Proper Names (Nice, Begin)	63	97
Roman Numerals (III)	69	95
Fractions/Dates (5/16)	71	94
Years/Quantifiers (1750)	66	93
Abbreviations (St., Dr.)	75	98
AVERAGE	68	96

4 Conclusion

The proposed algorithm offers considerable advantages for lexical ambiguity resolution in speech synthesis. It combines the strengths of N-gram taggers and Bayesian classifiers, utilizing both local syntactic patterns and wide-context lexical associations. By basing its classification on only the single best piece of evidence found in a target context rather than a combination of all available evidence, it allows the inclusion of multiple highly non-independent evidence sources (such as a word's lemma, part of speech, and inflected form) without modeling of the complex statistical dependencies. Furthermore, the decision list it generates is easily interpretable (unlike the output of many classification algorithms), highly efficient, and can be readily modified by hand as desired. The algorithm is able to incorporate a rich variety of evidence types and is easily applied to new domains.

[4]Pronunciation labels in the training and test data were hand tagged, using an earlier class-based sense disambiguator based on Roget's Thesaurus [6] for initial assignments.

References

[1] Brieman, L., J. Friedman, R. Olshen, and C. Stone, *Classification and Regression Trees*, Wadsworth & Brooks, Monterrey CA, 1984.

[2] Church, K.W., "A Stochastic Parts Program and Noun Phrase Parser for Unrestricted Text," in *Proceedings of the Second Conference on Applied Natural Language Processing, ACL*, 136-143, 1988.

[3] Gale, W., K. Church, and D. Yarowsky, "A Method for Disambiguating Word Senses in a Large Corpus," *Computers and the Humanities*, 26, 415-439, 1992.

[4] Rivest, R. L., "Learning Decision Lists," in *Machine Learning*, 2, 229-246, 1987.

[5] Sproat, R., J. Hirschberg and D. Yarowsky "A Corpus-based Synthesizer," in *Proceedings, International Conference on Spoken Language Processing*, Banff, 1992.

[6] Yarowsky, David "Word-Sense Disambiguation Using Statistical Models of Roget's Categories Trained on Large Corpora," in *Proceedings, COLING-92*, Nantes, 1992.

[7] Yarowsky, David, "One Sense Per Collocation," in *Proceedings, ARPA Human Language Technology Workshop*, 1993.

[8] Yarowsky, David, "Decision Lists for Lexical Ambiguity Resolution: Application to Accent Restoration in Spanish and French," in *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, Las Cruces, NM, 1994.