Improved pos tagging for text-to-speech synthesis

Conference Paper *in* Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on · June 2011

DOI: 10.1109/ICASSP.2011.5947575 · Source: IEEE Xplore

2 authors, including:

Jerome Bellegarda
Apple Inc.
126 PUBLICATIONS **2,359** CITATIONS

SEE PROFILE

IMPROVED POS TAGGING FOR TEXT–TO–SPEECH SYNTHESIS

Ming Sun[1] *and Jerome R. Bellegarda*[2]

[1]Center for Language & Speech Processing, Johns Hopkins University, Baltimore, MD 21218, USA
[2]Speech & Language Technologies, Apple Inc., Cupertino, CA 95014, USA

ABSTRACT

One of the fundamental building blocks of text processing for text-to-speech (TTS) synthesis is the assignment of a part-of-speech (POS) tag to each input word. POS tags are heavily relied upon for downstream natural language analysis and prosody rendering. Conventional TTS POS tagging tends to resort to detailed hand-crafted rules that can accommodate TTS specificities such as pertinent prosodic features, while mainstream tagging increasingly relies on data-driven statistical models trained on large but fairly generic corpora. This paper proposes a new strategy, *hybrid POS tagging*, which integrates these two approaches in order to achieve higher tagging accuracy. The resulting framework combines the TTS-specific advantage of rule-based tagging with the inherent robustness of broadly-trained statistical tagging. Empirical evidence underscores the viability of this framework for improving TTS quality, e.g., in regard to phrase boundary placement and homograph selection.

Index Terms: Speech synthesis, text processing, syntactic analysis, part-of-speech disambiguation, statistical/rule-based/hybrid tagging

1. INTRODUCTION

For a given input text, text-to-speech (TTS) synthesis performs a sequence of two distinct operations [1]. The first one is *text analysis*, whose role is to convert the input into an appropriate symbolic linguistic representation, such as a phone sequence annotated with pertinent prosody information. Then comes *signal generation*, which uses this representation as a guide to assemble suitable elementary units from some training inventory, either directly (as in unit selection synthesis [2]), or indirectly (for example, via trained statistical models like HMMs [3]). We focus here on the first operation: text processing significantly affects the level of quality that can be achieved by the overall TTS system, because any mistake introduced at that stage cannot be compensated for later on.

This operation relies on a number of natural language sub-tasks, ranging from text normalization to semantic parsing [4]. One of the fundamental building blocks is a shallow syntactic analysis of each sentence, in order to assign a suitable part-of-speech (POS) tag to each tokenized word, a task known as *POS tagging*. POS tags augment the information contained within words by explicitly indicating some of the structure inherent in language. Their accuracy is therefore critical to downstream sub-tasks, including chunking and semantic role labeling [5]. This in turn affects proper placement of prosodic markers such as accent types and phrase boundaries, which greatly influences how natural synthetic speech sounds. It is therefore important to make sure POS tagging works well in the context of TTS synthesis.

Historically, TTS POS tagging has predominantly been based on manually specified rules, typically hand-crafted using special-purpose tagsets taking into account TTS idiosyncrasies such as accent type and other prosodic features. Meanwhile, with the growing availability of natural language training resources in recent years, mainstream tagging has increasingly involved some form of data-driven statistical processing. State-of-the-art models based on conditional random fields (CRFs), for instance, are trained to identify the most likely sequence of tags for the observed set of words in a given sentence. These models rely on feature functions acting as marginal constraints to ensure that important characteristics of the empirical training distribution are reflected in the trained model. With well chosen functions covering sufficiently rich features of the training data, and given adequate initial conditions, CRF taggers can achieve a very high level of tag accuracy on general corpora [6]. But as is well known, the size and pertinence of the training data is critical to the quality of the resulting models.

There is, unfortunately, an inherent trade-off between size and pertinence. Standard corpora tend to be suitably extensive, but fairly generic in terms of supported tagset and associated annotation. Most of them use the default Penn Treebank POS tagset [7], which was of course not defined with TTS synthesis in mind. On the other hand, special-purpose databases used in voice building, for example, tend to be too small for the reliable estimation of CRF parameters. In addition, re-annotating a large corpus with a different tagset suitable for TTS is obviously not a cost-effective option. What seems to be needed is a solution that would combine the outcome of both approaches, so as to reap the benefits of both paradigms.

In this paper, we implement this strategy by introducing a framework called *hybrid POS tagging*. The paper is organized as follows. The next section provides some further motivation for an hybrid solution to POS tagging in the context of TTS synthesis. In Section 3, we present in detail our method for integrating rule-based and statistical taggers. Section 4 reports on experimental results obtained on a voice building database comprising about 5,000 sentences. Finally, in Section 5 we discuss a couple of illustrative examples representative of the type of behavior observed, which underscores the viability of hybrid POS tagging for improving TTS quality.

2. MOTIVATION

Given a natural language sentence comprising L words, POS tagging assigns to each observed word w_i some suitable part-of-speech p_i, $1 \leq i \leq L$. Representing the overall sequence of words by W and the corresponding sequence of POS by P, CRF taggers directly maximize the conditional probability $\Pr(P|W)$ over all possible POS sequences P. This is done via log-linear modeling of feature functions expressing important aspects of the empirical training distribution, as observed on a large annotated corpus. For example, in the sentence:

A chief judge heads the group. (1)

the POS sequence is ambiguous because most words have multiple POS.[1] In this simple case, however, feature functions covering suf-

[1]For example, "*A*" can nominally be either a noun (NNP) or a determiner (DT), "*chief*" can nominally be either a noun (NN) or an adjective (JJ), etc.

ficiently rich features of the context have no trouble performing the disambiguation.

The problem is that, for TTS synthesis, features commonly considered in mainstream statistical training are generally not sufficient. For example, in the sentence:

She is coming tomorrow, she is, she really is! (2)

the three instances of the word "*is*" would normally be assigned the same tag (VBZ). Yet, they are realized three different ways. The first instance is unaccented and reduced, the second is accented, and the third is unaccented but with full vowel quality. Any synthetic version not respecting these rendition patterns would not sound natural. It thus stands to reason that a TTS system would benefit from a POS assignment system which reflects such distinctions. At the very least, the first instance of "*is*" should be assigned a POS that usually carries no accent, such as auxiliary, and the second one a POS that usually carries an accent, such as (non-modal) verb.

This type of information is ordinarily encoded in additional tags, often leading to a tagset that differs from typical reference tagsets, and thereby creating a discrepancy with available annotated corpora used for training statistical taggers. This is why TTS POS tagging typically relies on hand-crafted rules developed from fairly limited databases. They can easily take into account the kind of distinctions exemplified in (2), including the case of the third instance of "*is*," which is clearly very specific to the TTS realm. On the other hand, they suffer from several potential drawbacks, including lack of portability, maintenance difficulties, and the risk of over-generalization from a small number of exemplars.

Hence the appeal of combining the two tagging techniques. Complementing a statistical tagger with a rule-based system solves many of the above problems: because the rules can now be focused on situations that are high-value for the application considered, in principle they can be fewer, simpler, and therefore more manageable. At the same time, generic training data can be leveraged to increase tagging robustness, without sacrificing specific requirements for the task at hand.

3. HYBRID TAGGING

In the hybrid system adopted, the two tagging approaches render independent assessments of each input word, one of which is then selected based on underlying conditions in order to produce the final POS tag for this word. There are therefore two main aspects to discuss: tagset unification, and the degree of consistency between the two assessments.

Tagset unification is done on the basis of plausibility: if a given tag from the statistical tagger can plausibly map onto a tag from the rule-based system, it is converted accordingly. Let $\mathcal{T}$ and $\mathcal{T}'$ be the respective tagsets for the rule-based and statistical systems. For each tag $t' \in \mathcal{T}'$, denote by $\mathcal{W}_{t'}$ the set of words which is assigned t' by the statistical tagger, and by $\mathcal{T}_{t'}$ the subset of $\mathcal{T}$ which includes all tags observed on $\mathcal{W}_{t'}$ when using the rule-based system. Then any tag $t \in \mathcal{T}_{t'}$ is allowable as the mapped rule-based tag version of the original statistical system tag t'. Conversion rules thus nominally include all of one-to-one, one-to-many, and many-to-one mappings. In practice, ties are resolved on the basis of frequency.

If after unification the two tagging techniques agree on a common tag, the final POS tag is taken to be that common tag. Otherwise, two cases need to be distinguished, depending on the behavior of the rule-based system.

If no suitable rule is found to apply to the input context, traditional tagging backs off to some default tag (generally the most frequent tag for the word considered). This typically forces an over-generalization, which in our experience is the source of most errors in rule-based systems. In this situation the associated assessment should not be relied upon. Thus, the final POS tag is taken to be the (statistical) CRF tag.

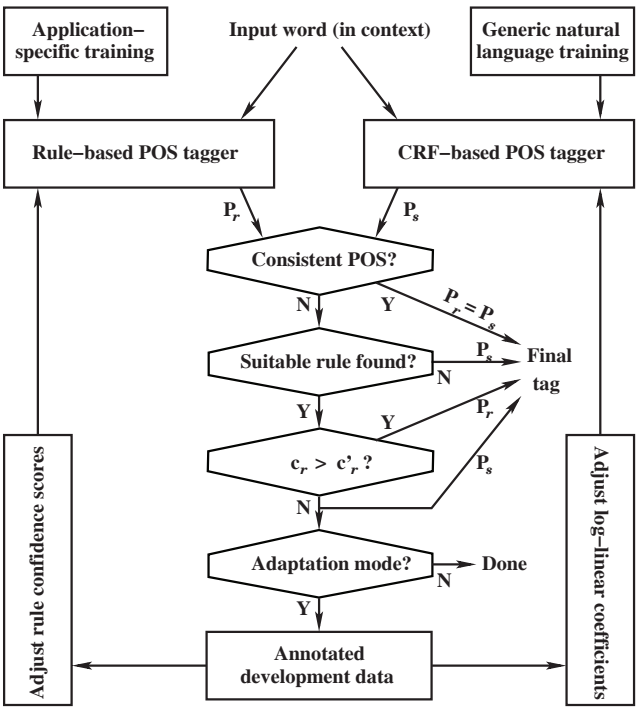

Fig. 1. *Combining CRF– and Rule–Based POS Tagging.*

If a suitable rule is found to apply to the input context, the end result is a disagreement between rule-based and statistical assessments. We resolve the ensuing dilemma by leveraging the confidence scores assigned to each rule.

Using some held-out development data, we collect all instances where the two tagging approaches disagree. For each rule r, we then tabulate the specific instances where each tagger was right and wrong, and compute two confidence scores as follows:

$$c_r = \frac{n_{r,i}}{n_{r,i} + n_{r,j}}, \quad c'_r = \frac{n'_{r,i}}{n'_{r,i} + n'_{r,j}}, \tag{3}$$

where $n_{r,i}$ and $n_{r,j}$ denote the number of instances for which the rule-based system was observed to be right and wrong, respectively, and $n'_{r,i}$ and $n'_{r,j}$ are analogous definitions for the statistical tagger.

From this information, it thus becomes possible to rank the rules in decreasing order of confidence scores c_r. Clearly, any rule with a confidence score $c_r \leq c'_r$ is obviously suspect, while any rule with a confidence score $c_r > c'_r$ exhibits a reasonable degree of reliability. During tagging, the final POS tag is then taken to be the rule tag if $c_r > c'_r$, and the CRF tag otherwise.

If desired, this information can then be fed back to the scoring mechanism in order to adjust the rule confidence scores for later reference. Likewise, it can be fed back to the CRF training in order to adjust the CRF parameters. Note that this adaptation process can be either supervised (if a human is in the loop to check the tag produced) or unsupervised (if no human feedback is requested). The complete decision algorithm is illustrated in Fig. 1.

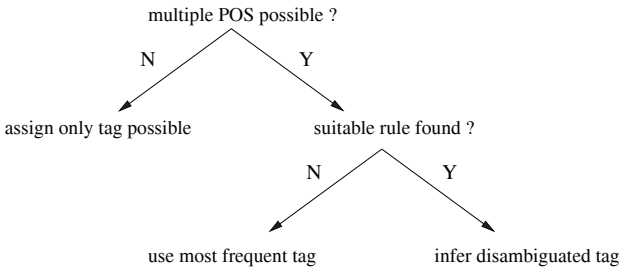

Fig. 2. *Data Flow in TTS Tagger.*

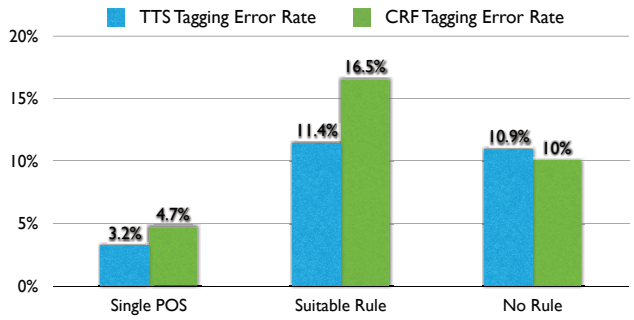

Fig. 3. *Tagging Error Rates for Three Ambiguity Levels.*

4. EXPERIMENTAL RESULTS

4.1. Experimental Setup

For evaluation purposes, the two components of the hybrid approach are as follows. The rule-based system is the tagger used in Apple's MacOS X 10.6 TTS engine, comprising about 170 rules. We will refer to it as the "TTS tagger." The statistical system is a tagger using the CRF paradigm, soon to be released as part of the "Appkit" portion of Apple's Cocoa API. We will refer to it as the "CRF tagger." Its nominal performance is roughly comparable to that of a typical POS tagger trained on the Wall Street Journal (WSJ) portion of the Penn Treebank, such as X.-H. Phan's *CRFTagger* [8].[2]

The CRF tagger relies on the standard Penn Treebank tagset [7], while the TTS tagger uses a different, TTS-centric tagset supporting a number of prosodic features. Tagset unification is thus done as described earlier. We map the Penn Treebank tagset into the TTS-centric tagset on the basis of either one-to-one mappings (e.g., RB → Adv), one-to-many mappings (e.g. VB → Verb|Vbe|Vhave), or many-to-one mapping (e.g., JJ|JJR|JJS → Adj). This allows for proper assimilation of distinctions that are immaterial to TTS, such as the difference between base, comparative, and superlative forms of adjective. At the same time, it preserves distinctions that are useful in TTS, such as the difference between a generic verb and any form of "to be" or "to have"—cf. (2).

Special care is warranted for words containing an apostrophe or an hyphen. The TTS tagger assigns a single POS to words with an apostrophe, while the CRF tagger breaks them into two parts and assigns a separate POS to each part. In this case, t' is taken to be the sequence of these two CRF tags. For example, in the phrase "*since the late eighteen hundred's*," the CRF tagger assigns the tag CD to "*hundred*"and the tag POS to "*'s*." This sequence is mapped into the single TTS tag Noun.

Words containing an hyphen are also handled in a special way. Both TTS and CRF taggers currently break them up into constituent parts and assign a separate POS to each part. This is not always desirable, as it sometimes leads to the wrong analysis. For example, in the phrase "*a gravity-defying stunt*," it is rather meaningless to break up the adjective "*gravity-defying*" into a sequence of, say, a noun and a present participle. In such cases, we thus use a handful of specific mapping rules to directly map the sequence of CRF tags into a single POS tag for the hyphenated word.

All systems were evaluated on a voice building database comprising 5,071 sentences (for a total of 57,013 words), representing a wide variety of styles and domains. On this test data, tagging error rate was measured to be 7.9% for the TTS tagger and (after tagset conversion) 10.1% for the CRF tagger. Note that the relatively poor performance of the statistical system compared to typical results reported in the literature is likely due to the rather severe mismatch between training and testing conditions.

[2]The early "vanilla" version considered here was generically trained, and, in particular, did not yet incorporate the kind of very rich features advocated in [9].

4.2. Ambiguity Levels

While the TTS tagger seems to nominally outperform the CRF tagger, the point is that numerous cases may exist for which the CRF tagger entails a higher tagging accuracy. To gain insights into the matter, it helps to consider the data flow of the TTS tagger, shown in Fig. 2. For any POS assignment, there are three possible outcomes:

- the given input word does not have multiple POS, in which case the only possible tag is assigned (left-most path);
- the word has multiple POS, but a suitable rule exists to resolve the ambiguity, in which case the appropriate tag is assigned after disambiguation (right-most path);
- no rule "fires" for the context at hand, in which case the TTS tagger must resort to a default tag (generally obtained from frequency information).

For the last outcome in particular, there is clearly room for improvement, as merely selecting the most frequent tag is unlikely to be the optimal course of action in all situations.

The three levels of ambiguity exemplified above can be illustrated on the following sentence:

It is now an endangered species. (4)

As the word "*species*" is normally a noun, a single POS is available, and POS assignment ensues as an instance of the first outcome. On the other hand, for the word "*endangered*" multiple POS exist (several verbal forms as well as adjective); however, one of the disambiguation rules fires, leading to the correct POS assignment as an instance of the second outcome. Finally, "*now*" also has multiple POS, but no disambiguation rules applies in the current version of the TTS tagger, so POS assignment follows as an instance of the third outcome. Note that, in this simple case, it happens to be correct, since "*now*" is most frequently an adverb.

For the test data considered, 41% of the tokenized words have a single POS available (typically the most likely to be associated with each word, excluding somewhat pathological cases), 34% have multiple POS but can potentially be disambiguated by an existing rule, and the rest (25%) receive a default tag. Categorizing the results in terms of these three ambiguity levels, the tagging error rates of the two taggers are shown in Fig. 3. The single POS class does not quite lead to perfect accuracy, because the tag assigned may still be incorrect in the specific context considered; in the case of the CRF tagger, this is further compounded by any error introduced during tag

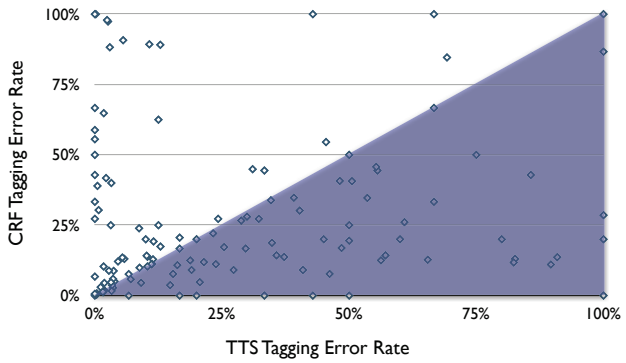

Fig. 4. *Comparative Error Rate Positioning for Each Rule.*

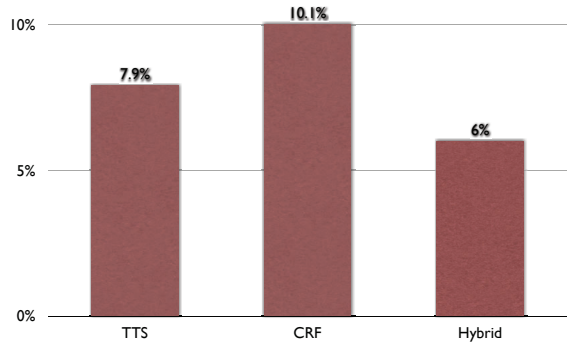

Fig. 5. *Overall Tagging Error Rate for All Three Taggers.*

conversion. When a suitable rule does exist, the TTS tagger substantially outperforms the CRF tagger, perhaps because most rules were carefully written to account for idiosyncratic linguistic/prosodic phenomena. For situations not covered by any rule, however, the CRF tagger has a small advantage, presumably because it can more reliably leverage events seen in its (large) training data.

4.3. Complementarity Analysis

Fig. 3 confirms that TTS/CRF integration should start by eliminating the TTS default tag from consideration, to capitalize on CRF's advantage in the "No Rule" case. Next, it seems unlikely that all rules would be performing uniformly well. Hence the interest to examine each rule individually to estimate its stand-alone accuracy. Basically, if a rule is found to satisfy $c_r \leq c'_r$, there is little justification in keeping it around in the hybrid tagger.

Fig. 4 shows the tagging error rates for all rules present in the TTS tagger, in terms of their comparative positioning with respect to the two taggers. (On Fig. 4, each blue diamond represents a particular rule.) For all points below the diagonal (blue area), the CRF tagger has a lower tagging error rate. Thus, it appears that approximately half the rules could potentially be eliminated from consideration by the hybrid tagger. Taking other criteria into account (such as firing frequency), 29 of those rules were actually eliminated.

Fig. 5 shows the final overall tagging error rate (6.0%) for the resulting hybrid tagger, compared to the original 7.9% and 10.1% figures obtained with the TTS and CRF tagger, respectively. This level of performance corresponds to a relative reduction of 24% compared to the TTS tagger and 41% compared to the CRF tagger.

5. ILLUSTRATIVE EXAMPLES

The POS error rate reduction achieved by the hybrid tagger improves the quality of TTS synthesis in several different ways. Two examples are illustrated below.

The placement of intermediate phrase boundaries is important to get right in speech synthesis. Consider the sentence:

This development thrills me for so many reasons. (5)

Because the original TTS tagger determines the word "*thrills*" to be a noun, the synthesizer incorrectly puts a boundary after "*thrills*" (cf. associated file "`thrills_org.aiff`"). On the other hand, the hybrid tagger derives the correct tag for "*thrills*," thus the boundary is moved immediately before the verb "*thrills*," which sounds more natural (cf. "`thrills_new.aiff`").

In the same vein, many words have different pronunciation based on different POS. For example, the word "*sow*" can be either a noun (pronounced [`sau`]) or a verb (pronounced [`so`]). Consider the sentence:

A farmer might feed his sow in the morning,
and sow a field of wheat in the afternoon. (6)

The original TTS tagger gets the wrong POS tag for the second "*sow*" and thus the synthesizer pronounces it incorrectly (cf. "`sow_org.aiff`"). On the other hand, the hybrid tagger derives the correct tag for it, which leads to the correct TTS synthesis (cf. "`sow_new.aiff`").

6. CONCLUSION

We have presented a hybrid POS tagging strategy which integrates a relatively small number of detailed hand-crafted rules with data-driven statistical tagging based on CRF. The resulting framework combines the application-specific advantage of rule-based tagging with the inherent robustness of generic CRF tagging in order to deliver "best of both worlds"-type accuracy. Evaluations on a 5,000-sentence voice building corpus show that hybrid POS tagging reduces the tagging error rate by 24%, leading to a number of perceivable improvements in TTS quality, e.g., in regard to phrase boundary placement and homograph selection. This bodes well for applying an analogous strategy to other natural language applications where high-accuracy tagging is desirable, such as grammar checking, spelling correction, machine translation, text completion, etc.

7. REFERENCES

[1] J.P.H. van Santen, R.W. Sproat, J.P. Olive, and J. Hirschberg, *Progress in Speech Synthesis*, New York, NY: Springer, 1997.

[2] A. Hunt and A. Black, "Unit Selection in a Concatenative Speech Synthesis System Using Large Speech Database," in *Proc. ICASSP*, Atlanta, GA, pp. 373–376, 1996.

[3] K. Tokuda, T. Yoshimura, T. Masuko, T. Kobayashi, and T. Kitamura, "Speech Parameter Generation Algorithms for HMM–Based Speech Synthesis," *Proc. ICASSP*, Istambul, Turkey, pp. 1315–1318, 2000.

[4] D. Jurafsky and J.H. Martin, *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*, 2nd Edition, Upper Saddle River, NJ: Prentice Hall, 2008.

[5] C.D. Manning and H. Schütze, *Foundations of Statistical Natural Language Processing*, Cambridge, MA: MIT Press, 1999.

[6] J. Lafferty, A. McCallum, and F. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data", in *Proc. 18th Int. Conf. Machine Learning (ICML 2001)*, Williamstown, MA, pp. 282–289, June 2001.

[7] M.P. Marcus, B. Santorini, and M.A. Marcinkiewicz, "Building a Large Annotated Corpus of English: the Penn Treebank," *Computational Linguistics*, Vol. 19, No. 2, pp. 313–330, 1993.

[8] X.-H. Phan, "CRFTagger: CRF English POS Tagger," `http://crftagger.sourceforge.net/`, 2006.

[9] K. Toutanova, D. Klein, C. Manning, and Y. Singer, "Feature–Rich Part–of–Speech Tagging with a Cyclic Dependency Network," in *Proc. HLT–NAACL*, Edmonton, Canada, pp. 252–259, May 2003.