# Adding Context Information to Part Of Speech Tagging for Dialogues

Sandra Kübler, Matthias Scheutz, Eric Baucom, Ross Israel

Indiana University
Bloomington, IN, USA
{skuebler,mscheutz,eabaucom,raisrael}@indiana.edu

**Abstract**

Part-of-speech (POS) tagging for English is often considered a solved problem, with accuracies for POS tagging the Penn Treebank of around 97%. However, POS tagging generally assumes that there is a large in-domain training set available, and that the domain is carefully edited written language. We investigate the performance of Markov model and maximum entropy POS taggers given a small data set of spontaneous dialogues in a collaborative search task. We investigate whether adding information about the speaker or about the dialogue move of the sentence can improve results. Our experiments show that especially the dialogue move information increases accuracy, but the information must be provided in a way that does not cause data sparseness issues. Our best results of 96.55% were reached by an extension of the maximum entropy tagger that uses the dialogue information as additional features in classification.

## 1 Introduction

Part-of-speech (POS) tagging for English is often considered a solved problem. There are well established approaches such as Markov model trigram taggers [1], maximum entropy taggers [10], or Support Vector Machine based taggers [7], and accuracy is around 97%. However, most experiments in POS tagging for English have concentrated on data from the Penn Treebank [9], i.e., based on a well defined genre (financial news) of carefully edited language and with a large training set. In this paper, we investigate POS tagging for a small corpus of spontaneous dialogues, CReST [6], based on an experiment in which humans perform a cooperative, remote search task. This corpus challenges both assumptions that are made when using the Penn Treebank: We have a very limited data set, and since the corpus is based on spontaneous dialogues, the language is more casual than in the Penn Treebank. I.e. this data set exhibits all the characteristics of spontaneous speech, including hesitations, false starts, corrections, and word replacement. Under these

conditions, POS taggers do not reach state-of-the-art results. For example, TnT, a Markov model POS tagger reaches 96.7% accuracy on the Penn Treebank [1] but approximately 2 percent points less, 94.8% when trained and tested on the CReST corpus. If TnT is trained on the Penn Treebank and tested on the CReST corpus, it reaches an accuracy of only 85.5%. Consequently, there are significant differences between the Penn Treebank and CReST, which cannot be counterbalanced by the shorter and less complex sentences in CReST, which should be easier to analyze.

In interpreting these numbers, we have to take into consideration that a wrong choice in the POS tag of an ambiguous word will negatively affect following analysis steps such as syntactic parsing. One typical error in our experiments is the confusion of adverbs and adjectives. If an adjective is erroneously tagged as an adverb, then the parser may be forced to resort to a different subcategorization frame for a verb, thus changing the analysis for the complete sentence. This means that even a 2% drop in POS tagging accuracy will result in a serious decrease in performance of a parser that uses the POS tagged text as input.

In order to improve POS tagging results for small training corpora, we investigated whether adding information about the context improves tagging accuracy. The context information that we included was dialogue model information, and more specifically information about the speaker (director or searcher), information about the dialogue move, and a combination of both. Our hypothesis is that the dialogue model influences the types of sentences uttered at a certain point in the dialogue, and consequently also the types of POS tags. We assume that since dialogue moves tend to be associated with certain syntactic structures, they would also help predict certain sequences of tags. For instance, if the move is INSTRUCT, the sentence is more likely to be a command, which informs the POS sequence, especially at the beginning of the sentence. We also assume that speaker information would help predict the tag sequence because different speaker roles tend to produce different syntax structures, especially when the roles are as distinct as director and searcher, which was the case in our corpus. The searcher, for example, may be more likely to ask questions, which have different POS sequences. Our goal in comparing different POS taggers and different ways of integrating the information into the POS tagging process was to find the optimal approach for integrating additional context information into the POS tagging process.

Our results show that adding the additional information is indeed helpful, but only if it is integrated as additional features and not in the POS tags themselves. We also show that maximum entropy tagging is better suited for integrating new types of information.

The remainder of the paper is structured as follows: In section 2, we discuss related approaches, and in section 3, we discuss the corpus in more detail as well as the POS taggers used for the experiments. Section 4 outlines our experimental methodology. We present our results in section 5, and we finish with our conclusion and future work in section 6.

## 2 Related Work

There is a vast literature on many facets of POS tagging. For the purpose of the research presented here, we concentrate on POS tagging approaches that go beyond using 1-2 words on either side of the focus word as information in deciding the POS tag of the focus word. To our knowledge, the only statistical POS tagging approach that does use additional features is the one based on maximum entropy models. Maximum entropy taggers allow for a rich, linguistically motivated feature set to be used in a statistical framework. Accordingly, there have been a number of maximum entropy POS taggers developed in an attempt to further improve upon the accuracy that can be achieved by other approaches.

Ratnaparkhi [10] describes a maximum entropy approach to POS tagging. The tagger learns a log-linear conditional probability model from a training corpus of tagged text using a maximum entropy classifier. Along with contextual features looking at the surrounding words and tags, there are a number of features based on the form of the word, including the nature of affixes and the inclusion of hyphens, apostrophes, numbers, and capital letters. Ratnaparkhi reports 96.6% accuracy on unseen data from the Wall Street Journal.

Toutanova and Manning [13], also working with a maximum entropy tagger, used the model laid out by Ratnaparkhi [10] as a starting point, and raised the effectiveness of the tagger to deal with unknown words, in particular. They added a set of features designed to help identify proper nouns, another set to disambiguate verb forms, and a set to disambiguate participles, adverbs, and prepositions more accurately. By adding these features and excluding some that were used by Ratnaparkhi, Toutanova and Manning were able to achieve higher accuracy overall and specifically for unknown words on the same test set from the Wall Street Journal.

Inspired by the work of Ratnaparkhi and Toutanova and Manning, Denis and Sagot [5] developed another implementation of a maximum entropy tagger, MElt, that they applied to the task of POS tagging French. The authors developed a superset of features that combined those utilized previously. The full set of features is shown in Table 1. Denis and Sagot altered the algorithm by Ratnaparkhi by lifting the restriction that so called lexical features, i.e. features that examine the composition of words, should only apply to rare words. They also added an external lexical resource, *Lefff* [11], to be used in concert with the dictionary learned from the training data. MElt is described further in the following section.

## 3 Corpus and POS Taggers

### 3.1 The CReST Corpus

The CReST corpus [6] is a corpus of natural language dialogues obtained from humans performing a cooperative, remote search task in which one person outside the search environment (director) directed a person inside the environment (searcher). The director guided the searcher through the search environment, for

| **Lexical features** | |
|---|---|
| $w_i = X$ | & $t_i = T$ |
| Prefix of $w_i = P, \|P\| < 5$ | & $t_i = T$ |
| Suffix of $w_i = S, \|S\| < 5$ | & $t_i = T$ |
| $w_i$ contains a number | & $t_i = T$ |
| $w_i$ contains a hyphen | & $t_i = T$ |
| $w_i$ contains an uppercase letter | & $t_i = T$ |
| $w_i$ contains only uppercase letters | & $t_i = T$ |
| $w_i$ does not start a sentence and contains an uppercase letter | & $t_i = T$ |
| **Contextual features** | |
| $t_{i-1} = X$ | & $t_i = T$ |
| $t_{i-2}t_{i-1} = XY$ | & $t_i = T$ |
| $w_{i+j} = X, j \in -2, -1, 1, 2$ | & $t_i = T$ |

Table 1: Denis and Sagot's MElt features.

which the director had a map, in order to find different colored boxes, enter them on the map, and place blocks in them. The director was fitted with a free-head eyetracker, and he was recorded by a microphone positioned between the director and the telephone's speaker. The searcher wore a helmet with a cordless phone and a light-weight digital video camera that recorded his or her movement through the environment as viewed from his or her perspective and provided a second audio recording of the spoken dialogue.

The multi-modal corpus consists of seven dialogues. The text highlights the differences between formal written and naturally occurring language, as it is rife with directives, disfluencies, corrections, ungrammatical sentences, wrong-word substitutions, and various other constructions that are missing from written text corpora. In total, there are 11 317 words in 1 977 sentences.

The corpus contains the speech signals as well as transcriptions of the dialogues, which are additionally annotated for dialogue structure, disfluencies, and for syntax. The syntactic annotation comprises POS annotation, Penn Treebank [9] style constituent annotations, as well as dependency annotations based on the dependencies of *pennconverter* [8].

## 3.2 Annotation

On the dialogue level, the corpus was annotated for dialogue structure and for disfluencies. Utterances were divided into separate dialogue moves, based on the classification developed by Carletta et al. [2] for coding task-oriented dialogues. Their scheme views utterances as moves in a conversational game and classifies utterances into three basic move categories: *Initiation*, *Response*, and *Ready*. *Initiation* is further divided into INSTRUCT, EXPLAIN, QUERY-YN, QUERY-W, CHECK, and ALIGN. The category *Response* includes ACKNOWLEDGE, replies to wh-questions REPLY-WH, and `yes` or `no` replies REPLY-Y, REPLY-N.

| | | | |
|---|---|---|---|
| yeah | AP | you | PRP |
| let | VBI | ’re | VBP |
| ’s | PRP | gonna | VBG+TO |
| do | VB | find | VB |
| that | DDT | a | DT |
| yeah | UH | pink | JJ |
| | | box | NN |

Figure 1: Two examples with POS annotation.

The POS annotation is based on the Penn Treebank POS tagset [12], with a small number of new POS tags added to describe typical characteristics of spoken language:

- **AP** for adverbs that serve for answering questions, such as `yes`, `no`, or `right`.
- **DDT** for substituting demonstratives, such as in `that is correct`.
- **VBI** for imperatives, such as `turn left`.
- **XY** for non-words or interrupted words.

The first sentence in Figure 1 shows an example of a sentence with three new POS tags.

Another modification of the tagset concerns informal contractions such as in `you ’re gonna wanna turn to the right?`, which are kept as single words. As a consequence, they are assigned combinations of tags, such as **VBG+TO**. The second sentence in Figure 1 shows an example of such a contraction.

### 3.3 POS Taggers

We used two different POS tagging approaches, Markov models, and maximum entropy models. In the following, we give a short overview of the individual implementations and their characteristics.

**TnT.** TnT [1] is a trigram Markov model POS tagger with state-of-the-art treatment of unknown words. It can be trained on new data sets, and the implementation allows setting parameters such as the order of the Markov model, but it is impossible to add new types of data because the source code for the POS tagger is not available.

**IncT.** In order to incorporate new types of information, we used our own re-implementation of an incremental trigram Markov model POS tagger. The trigram model is interpolated with unigram and bigram models, using $\lambda$ values taken from TnT's optimization. For handling unknown words, IncT uses a simpler version of TnT's suffix trie in combination with Chen-Goodman smoothing [3].

**MElt.** For a maximum entropy tagger, we chose the Maximum-Entropy Lexicon-Enriched Tagger, MElt [5]. MElt is a conditional sequence maximum entropy POS tagger that uses a set of lexical and context features, which are a superset of the features used by Ratnaparkhi [10] and Toutanova and Manning [13]. The features are handled by the MegaM maximum entropy package [4]. The implementation, including the source code, is available from sourceforge.

**MElt+.** In order to integrate new types of information, we modified the MElt source code to add any features that accompany a sentence in a comment line at the beginning of the sentence. The modification only adds these new features so that there is no change in performance or accuracy when no features are added.

# 4 Experiments

We used the seven dialogues as folds in a 7-fold cross-validation. Evaluation was performed on the concatenation of the test data sets, i.e. on the whole data set. As a baseline, we used all POS taggers without modification. This means that MElt and MElt+ are identical, and we report only 3 results. Then we added the dialogue information about the speaker and the dialogue move assigned to the sentence. Both types of information were extracted from the multi-modal corpus annotation. In a first experiment, we added this information to the POS tags, thus creating complex POS tags. This approach has the advantage of not requiring any modification of the POS taggers. In a second experiment, we added the new type of features directly to the algorithms. For these experiments, we experimented with adding the speaker information, the dialogue move information, and a combination of both types of information.

The evaluation was performed on POS tags only; in the experiments using the complex tags, we used the complex tags for training and testing, but for evaluation, we stripped off the additional information and evaluated on the POS tags only. One reason for this procedure is that we needed to ensure comparability between experiments. The more important reason is that we are not interested in how accurately the POS tagger can predict the additional information but rather in whether the additional information is useful for tagging and whether it can be successfully built into the POS tagging process.

### 4.1 Complex POS Tags

The simplest approach to adding the new information is to add it to the POS tags themselves, thus creating more complex tags. For example, given the word-tag combination `left/VBN` spoken by the director during a question move, we create the complex tag `left/VBN_director` when only speaker information is added, `left/VBN_question` with dialogue move information, and `left/VBN_director_question` with both types of information.

Adding the new information to the POS tags increases the size of the tagset and therefore also the risk of data sparseness: While the original POS tagset contains 38 different POS tags, adding the speaker information increases the tagset to 74 tags, and adding the dialogue moves results in a large tagset of 515 different tags. Adding a combination of both types of information results in a tagset of 772 tags. It is obvious that the latter two tagsets can very easily result in data sparseness problems, given that we only have small corpus of 11 317 words.

### 4.2 Modified Algorithms

In order to avoid data sparseness issues with the extended data sets, we pursued a second approach in which we integrated the new information into the algorithms directly. For MElt, this conversion to MElt+ was relatively simple: MElt uses a maximum entropy classifier in the background. Thus, for each word, an instance with independent features is passed to the classifier, which then makes the decision which POS tag should be assigned to the word based on the features. Our modification of the algorithm consists of passing the new types of information to the classifier as additional features.

For IncT, the modification was more extensive. To integrate the new information into the Markov model, we replaced the standard sentence boundary marker by a set of such markers, which model the additional features. Thus, in the training phase, counts were tabulated of how often certain POS tags occurred at the beginning and end of the sentences, in the context of certain dialogue moves and/or speaker types. For the combination of both types of information, we first used a solution in which we combined the labels into a single tag, e.g. `QYN_Searcher` for a yes/no question uttered by the searcher. Since this led to data sparseness issues, we then modified the approach so that the first sentence boundary marker at the beginning of the sentence represents the speaker, and the second sentence boundary the dialogue move. For the sentence boundary marker, we chose again the dialogue move. Thus the trigrams extracted from the INSTRUCT sentence `turn/VBI left/RB` uttered by the director are shown in Figure 2. The sentence boundary markers start with a $ sign.

| | | |
|---|---|---|
| $Director | $INSTRUCT | VBI |
| $INSTRUCT | VBI | RB |
| VBI | RB | $INSTRUCT |

Figure 2: The trigrams extracted from the sentence `turn/VBI left/RB`.

# 5 Results

The results of the experiments described above are shown in Table 2. The first baselines, in which the POS taggers were trained on the Penn Treebank and tested on CReST, show that both the trigram and the maximum entropy tagger do not perform well out of domain; TnT reached 85.49%, and MElt, which relies more heavily on lexical features, reached 83.31%. Since the initial results were so low, we refrained from repeating this experiment with IncT. From these experiments, we can conclude that using an existing model trained out-of-domain does not provide useful results. When the taggers are trained on CReST in 7-fold CV, the baseline shows that although the taggers were trained on a small data set in the order of 9 700 words, they reached results that are only slightly lower than results reported on the Penn Treebank (Brants [1] reports an accuracy of 96.7% on this data set). On our data set, TnT reached 94.80% while MElt reached a slightly higher accuracy of 95.64%. Our own trigram Markov model tagger, IncT, reached an accuracy that is comparable to TnT's 94.50%. The slight difference can be explained by the taggers' different strategies for handling unknown words.

| | | TnT | IncT | MElt | MElt+ |
|---|---|---|---|---|---|
| Baseline | Trained on Penn | 85.49 | * | 83.31 | |
| | Trained on CReST | 94.80 | 94.50 | 95.64 | |
| Complex Tags | Dialogue Move | 94.42 | 89.28 | 94.70 | |
| | Speaker | 94.81 | 93.57 | 95.39 | |
| Modified Algorithm | Dialogue Move | * | 95.03 | * | 96.55 |
| | Speaker | * | 94.52 | * | 95.74 |
| | Dialogue Move & Speaker | * | 94.98 | * | 96.55 |

Table 2: Results of the POS tagging experiments

In the experiments reported as **Complex Tags**, we added the additional information to the POS tags, thus creating complex tags. A closer look at the table corroborates our assumption that such a procedure leads to data sparseness. All taggers performed worse than in the baseline experiments. The only exception is the experiment in which TnT was confronted with POS tags that contained speaker information. In this experiment, TnT reached a non-significant[1] improvement of 0.1 percent points over the baseline. Adding speaker information results in a smaller

[1]McNemar, $p < 0.001$.

loss of accuracy for all taggers than adding dialogue moves. The reason for this difference can be found in the potential increase in POS tags that adding dialogue moves causes. Speaker information consists of 2 labels, director and speaker. Thus it can maximally double the initial POS tagset of 38 tags, and almost does: We observed 74 out of the 76 possible tags. Adding dialogue moves, in contrast, supplies 47 new labels, which can maximally create $38 * 47 = 1\,786$ complex labels. Even though the actual number is considerably lower at 515, there is still an increase by more than a factor of 13. The fact that this major increase in tags only results in losses in accuracy of less than 1 percent points for TnT and MElt shows that the new information must provide useful information. However, while TnT and MElt suffered minimally, adding the dialogue move information for IncT resulted in a considerable loss of accuracy. The tagger only reached an accuracy of 89.28%, which is more than 5 percent points lower than the baseline accuracy.

Since both types of information result in lower accuracies, we refrained from adding both types simultaneously. This would have increased the size of the tagset even more and thus exacerbated the data sparseness problem. Instead, we investigated whether the information can be successfully integrated into the algorithms. The results of these experiments are reported as **Modified Algorithm**. Since it is not possible to use the original implementations for these experiments, we report results only for IncT and for MElt+.

A closer look at the results of these experiments shows that adding speaker information results in a non-significant improvement for both MElt+ and IncT. The error reduction for MElt+ is 2.4% and for IncT 0.3%. In contrast, adding dialogue move information results in a significant increase for both taggers, with an error reduction of 20.9% for MElt and 9.5% for IncT. This increase shows that dialogue information is more useful in POS tagging CReST than speaker information. Why speaker information is not more helpful is not immediately clear. However, the setup of the search scenario is such that both speakers need to collaborate to perform the tasks. This means that both speakers ask questions or give directions during the completion of the task. For example, the corpus contains 45 questions asked by the director and 89 questions asked by the searcher. However, there are large individual differences between the dialogues; in two dialogues, the director asks more questions than the searcher. Thus, knowing whether a word is part of a question or of an explanation is more useful than knowing which speaker uttered the sentence.

A very clear example where the dialogue moves provide useful information for POS tagging is the word `yeah`. This word is assigned the POS tag `AP` when it occurs in an answer to a yes/no question, i.e. when it is part of a REPLY-Y or REPLY-N move, and it is assigned the POS tag `UH` when it belongs to any other move. There is only one exception this rule: In the sentence `yeah let 's do that yeah`, the second `yeah` is tagged `UH` in spite of being in a REPLY-Y move. In addition, the confusion between `AP` and `UH` is the largest source of errors in our experiments. Adding speaker and dialogue move information results for this word in an error reduction of 91.1% for MElt+ and of 91.3% for IncT.

While adding individual information is either beneficial or slightly detrimental, the picture is much clearer in the experiment where both types of information are added: IncT has a minimal decrease in accuracy to 94.98% in comparison to adding only dialogue move information, MElt+ reaches the same accuracy as in the experiment with dialogue moves, 96.55%. In comparison to the in-domain baseline, however, IncT reaches an error reduction of 9.8%, and for MElt+, there is an error reduction of 20.9%.

This shows that both types of information are potentially useful for POS tagging dialogue data. However, the information must be integrated in a way in which a POS tagger can successfully use the information without encountering data sparseness. Since MElt+ adds both types as individual features, no data sparseness ensues. IncT, in contrast, must use a combination of both tags and thus cannot avoid data sparseness. This leads us to conclude that adding the additional information as sentence boundary markers is not viable. Instead, the information must be integrated into the transition probabilities.

## 6 Conclusion and Future Work

In this paper, we have shown that for dialogue data such as in the CReST corpus, adding information about the speaker and about the dialogue moves improves tagging results. Especially, dialogue move information provides valuable disambiguation information for words that can be ambiguous between different categories that primarily occur in certain dialogue moves. However, in order to avoid data sparseness, we had to provide the data not as part of complex POS tags but rather as information inside the POS tagging algorithm. The results show that adding features to a maximum entropy tagger results in higher accuracy than adding them as sentence boundary markers in a Markov model tagger.

For the future, we are planning to modify IncT, the incremental Markov model tagger, so that the calculation of the transition probabilities is not conditioned on the previous context words but also on the additional information. This modification will allow us to use the additional information in all decisions. In order to avoid data sparseness, we will also modify the interpolation model.

We are also planning on extending our experiments and integrating a classifier for dialogue moves. We do have a preliminary version, which reaches an accuracy of approximately $70\%$ by looking only at previous move information, with disregard of the words in the sentences. While this is a module with state-of-the-art accuracy for dialogue moves, it is likely that the error rate is still too high to have a positive influence on POS tagging.

## Acknowledgment

This work is based on research supported by the US Office of Naval Research (ONR) Grant #N00014-10-1-0140.

## References


[1] Thorsten Brants. Tnt - a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing (ANLP-2000)*, Seattle, WA, 2000.

[2] Jean Carletta, Stephen Isard, Amy Isard, Gwyneth Doherty-Sneddon, Jacqueline Kowtko, and Anne Anderson. The reliability of a dialogue structure coding scheme. *Computational Linguistics*, 23(1):13–31, 1997.

[3] Stanley Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. Technical Report TR-10-98, Harvard University, 1998.

[4] Hal Daumé III. Notes on CG and LM-BFGS optimization of logistic regression. Paper available at `http://pub.hal3.name#daume04cg-bfgs`, implementation available at `http://hal3.name/megam/`, August 2004.

[5] Pascal Denis and Benoît Sagot. Coupling an annotated corpus and a morphosyntactic lexicon for state-of-the-art POS tagging with less human effort. In *Proceedings of the Pacific Asia Conference on Language, Information and Computation (PACLIC 23)*, Hong Kong, China, 2009.

[6] Kathleen Eberhard, Hannele Nicholson, Sandra Kübler, Susan Gunderson, and Matthias Scheutz. The Indiana "Cooperative Remote Search Task" (CReST) Corpus. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC)*, Valetta, Malta, 2010.

[7] Jesús Giménez and Lluís Màrquez. SVMTool: A general POS tagger generator based on Support Vector Machines. In *Proceedings of the 4th International Conference on Language Resources and Evaluation (LREC)*, Lisbon, Portugal, 2004.

[8] Richard Johansson and Pierre Nugues. Extended constituent-to-dependency conversion for English. In *Proceedings of NODALIDA 2007*, Tartu, Estonia, 2007.

[9] Mitchell Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, 19(2):313–330, 1993.

[10] Adwait Ratnaparkhi. A maximum entropy model for part-of-speech tagging. In *Proceedings of the Empirical Methods in Natural Language Processing Conference (EMNLP)*, Philadelphia, PA, 1996.

[11] Benoît Sagot, Lionel Clément, Éric De La Clergerie, and Pierre Boullier. The Lefff 2 syntactic lexicon for French: Architecture, acquisition, use. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, pages 1348–1351, Genoa, Italy, 2006.

[12] Beatrice Santorini. Part-of-speech tagging guidelines for the Penn Treebank Project. Department of Computer and Information Science, University of Pennsylvania, 3rd Revision, 2nd Printing, 1990.

[13] Kristina Toutanova and Christopher D. Manning. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (EMNLP/VLC)*, Hong Kong, 2000.