A Comparison of Language Models for Dialog Act Segmentation of Meeting Transcripts

Jáchym Kolář

Department of Cybernetics at Faculty of Applied Sciences,
University of West Bohemia, Univerzitní 8, 306 14 Plzeň, Czech Republic
jachym@kky.zcu.cz

Abstract. This paper compares language modeling techniques for dialog act segmentation of multiparty meetings. The evaluation is twofold; we search for a convenient representation of textual information and an efficient modeling approach. The textual features capture word identities, parts-of-speech, and automatically induced classes. The models under examination include hidden event language models, maximum entropy, and BoosTexter. All presented methods are tested using both human-generated reference transcripts and automatic transcripts obtained from a state-of-the-art speech recognizer.

1 Introduction

Recent years have witnessed significant progress in the area of automatic speech recognition (ASR). Nowadays, large volumes of audio data can be transcribed automatically with reasonable accuracy. Although the application of automatic methods is extremely labor saving, raw automatic transcripts often do not have a form convenient for subsequent processing. The problem is that standard ASR systems output only a raw stream of words, leaving out important structural information such as locations of sentence or dialog act (DA) boundaries. Such locations are overt in standard text via punctuation and capitalization, but "hidden" in speech.

As proved by a number of studies, the absence of linguistic boundaries is confusing both for humans and computers. For example, Jones et al. showed that sentence breaks are critical for legibility of speech transcripts [1]. Likewise, missing sentence or DA boundaries cause significant problems to automatic downstream processes. Many natural language processing techniques (e.g., parsing, automatic summarization, information extraction and retrieval, machine translation) are typically trained on well-formatted input, such as text, and fail when dealing with unstructured streams of words. For instance, Kahn et al. reported a significant error reduction in parsing performance by using an automatic sentence boundary detection system [2].

This paper deals with automatic linguistic segmentation of multiparty meetings from the ICSI corpus [3]. The goal is to segment the meeting transcripts into meaningful utterance units. The target units herein are not defined as sentences but as DAs. Although the original manual transcripts of the ICSI corpus do contain punctuation, and thus also sentence boundaries, the punctuation is highly inconsistent. Transcribers were instructed to focus on transcribing words as quickly as possible; there was not a focus on consistency or conventions for marking punctuation. Hence, instead of using the

inconsistent first-pass punctuation, it was decided to employ special DA segmentation marks from the MRDA annotation project [4]. In this annotation pass, labelers carefully annotated both dialog acts and their boundaries, using using a set of segmentation conventions for the latter. For a given word sequence, the task of DA segmentation is to determine which inter-word boundaries correspond to a DA boundary. Each inter-word boundary is labeled as either a within-DA boundary or a boundary between two DAs.

There are two basic sources of information that can be used to solve the task: recognized words and prosody. Several different approaches relying on one or both of the information sources have been employed for sentence and DA segmentation [5–10]. In this paper, I focus on an effective utilization of the information contained in the recognized stream of words. Well-tuned language models (LMs) are not only important for applications where they are combined with a prosody model, but also for the applications in which we do not have access to, or cannot exploit, prosodic information.

The LM evaluation is twofold; I search both for a convenient representation of textual information and an efficient modeling approach. In terms of textual knowledge representation, I analyze contributions from word identities, parts-of-speech, and automatically induced word classes. In terms of statistical modeling, I explore three different approaches – hidden event language models, maximum entropy models, and boosting-based models. I test the methods using both reference human-generated transcripts and automatic transcripts obtained from a state-of-the-art speech recognizer. I also address the issue whether it is better to train the system on clean reference data or on data containing word recognition errors.

2 Method

2.1 Data and Experimental Setup

The ICSI meeting corpus contains approximately 72 hours of multichannel conversational English. The data were split into a training set (51 meetings, 539k words), a development set (11 meetings, 110k words), and a test set (11 meetings, 102k words). The test set contains unseen speakers, as well as speakers appearing in the training data as it is typical for the real world applications.

For model training and testing I used both human-generated reference transcripts and ASR output. Recognition results were obtained using the state-of-the-art SRI speech recognition system [11]. Word error rates for this difficult data are still quite high; the used ASR system performed at 38.2% (on the whole corpus). To generate the "reference" DA boundaries for the ASR words, the reference setup was aligned to the recognition output with the constraint that two aligned words could not occur further apart than a fixed time threshold. DA boundaries occupy 15.9% of inter-word boundaries in reference and 13.9% in automatic transcripts.

2.2 Textual Features

In this work, I do not only use simple word-based models, but also utilize textual information beyond word identities, as captured by word classes and part-of-speech tags.

I do not use chunking (or even full-parsing) features. Chunking features may slightly increase performance for well-structured speech such as broadcast news [12], but preliminary investigations showed that, because of poor chunking performance on meeting data, these features rather hurt DA segmentation accuracy on meeting speech. Hence, I did not use them in this work. The following sections describe individual groups of employed features.

Words Word features simply capture word identities around possible DA boundaries and represent a baseline for our experiments.

Automatically Induced Classes (AIC) In language modeling, we always have to deal with data sparseness. In some tasks, we may mitigate this problem by grouping words with similar properties into word classes. The grouping reduces the number of model parameters to be estimated during training. Automatically induced classes (AIC) are derived in a data-driven way. Data-driven methods typically perform a greedy search to find the best fitting class for each word given an objective function.

The clustering algorithm I used [13] minimizes perplexity of the induced class-based n-gram with respect to the provided word bigram counts. The DA boundary token was excluded from merging, however, its statistics still affected the clustering. The algorithm works as follows. Initially, each word is placed into its own class. Then, the classes are iteratively merged until the desired number of clusters is reached. The resulting classes are mutually exclusive, i.e., each word is only mapped into a single class. In every step of the algorithm, the overall perplexity is minimized by joining the pair of classes maximizing the mean mutual information of adjacent classes

$$I(c_1, c_2) = \sum_{c_1, c_2 \in C} P(c_1, c_2) \log \frac{P(c_1, c_2)}{P(c_1)P(c_2)} \tag{1}$$

A crucial parameter of the word clustering algorithm is the target number of classes. The optimal number was empirically estimated on development data by evaluating performance of models with a different granularity. I started with a 300-class model and then was gradually decreasing the number of classes by 25 in each iteration. The optimal number of classes was estimated as 100.

I also tested a model that mixes AICs and frequent words by excluding them from class merging. This approach can be viewed as a form of back off; we back off from words to classes for rare words but keep word identities for frequent words. I have tested various numbers of left out words in combination with individual class granularities, but have never achieved better results than for the 100 classes with no excluded words.

Parts-of-speech (POS) The AICs reflect word usage in our datasets, but do not form clusters with a clearly interpretable linguistic meaning. In contrast, part-of-speech (POS) tags describe grammatical features of words. The POS tags were obtained using the TnT tagger [14] which was tailored for conversational English. The tagger was trained using hand-labeled data from the Switchboard Treebank corpus. To achieve a better match

with speech recognition output used in testing, punctuation and capitalization information was removed before using the data for tagger training [12].

Same as for AICs, I also tested mixed models. In contrast with AICs, mixing of frequent words with POS of infrequent words yielded an improvement. The reason is that while the automatic clustering algorithm takes into account bigram counts containing the DA boundary token and thus is aware of strong DA boundary indicators, POS classes are purely grammatical. By keeping the frequent words we also keep some strong boundary indicators. Optimizing the model on the development data, I ended up with 500 most frequent words being kept and not replaced by POS tags.

2.3 Statistical Language Models

Hidden Event Language Models (HELM) In speech recognition as well as in a number of other language modeling tasks, the role of the language model is to predict the next word given the word history. In contrast, the goal of language modeling in our task is to estimate the probability that an event, such as DA boundary, occurs in the given word context. Because these events are not explicitly present in the speech signal, they are called *hidden*. The hidden event LMs (HELMs) [5] describe the joint probability of words and hidden events $P(W, E)$ in an HMM. In this case, the HMM hidden variable is the type of the event (including "no-event"). The states of the model correspond to word/event pairs and the observations to words.

The model is trained by explicitly including the DA boundary as a token in the vocabulary in an n-gram LM. I used trigram LMs with modified Kneser-Ney smoothing [15]. In addition, Witten-Bell smoothing was employed for unigrams in class-based models (both AIC and POS) since the training data for these models do not contain any unigram singletons necessary for the Kneser-Ney method. During testing, the model performs the forward-backward decoding to find the DA boundaries given the word sequence. An implementation of the HELM is available in the SRILM toolkit [16].

The HELM does not allow a direct combination of multiple knowledge sources. Thus, I trained a separate model for each data stream and combined the models using a linear interpolation with weights estimated on development data.

Maximum Entropy Models The above described HELM is a generative model. It means that during training, it does not directly maximize the posterior probabilities of the correct classes. On the other hand, Maximum Entropy (MaxEnt) [17] is a discriminative model which is trained to directly discriminate among the possible target classes. This setup avoids the mismatch between training and using the model in testing. MaxEnt framework also allows a natural combination of multiple knowledge sources within a single model, no additional model combination is necessary. MaxEnt belongs to the exponential (or log-linear) family of classifiers, i.e. the features are combined linearly, and then used as an exponent

$$P(y|x) = \frac{1}{Z(x)} \exp\left(\sum_i \alpha_i f_i(x, y)\right) \tag{2}$$

where $Z(x)$ is a normalization factor ensuring that $\sum_y p(y|x) = 1$.

An important feature of MaxEnt models is that they are prone to overfitting. To overcome this drawback, I have used smoothing with Gaussian priors that penalizes large weights. For all experiments with MaxEnt, I employed the `MegaM` toolkit.[1] For each feature group, the used features included all n-grams up to trigrams spanning across or neighboring with the inter-word boundary in question. I also added a binary feature indicating whether the word before the boundary is identical with the following word. This feature aims to capture word repetitions.

Boosting-based Models (BoosTexter) Boosting is an aggregating machine learning method combining many weak learning algorithms to produce an accurate classifier. Each weak classifier is built based on the outputs of previous classifiers, focusing on the samples that were formerly classified incorrectly; the algorithm generates weak classification rules by calling the weak learners repeatedly in series of rounds. This approach can generally be combined with any "weak" classifier. In this work, an algorithm called BoosTexter [18] was employed.

BoosTexter was initially designed for the task of text categorization, employment of this method for tasks related to DA segmentation was firstly presented in [9, 19]. The method combines weak classifiers having a basic form of one-level decision trees (stumps) using confidence-rated predictions. The test at the root of each tree can check for the presence or absence of an n-gram, or for a value of a continuous or categorical feature. Same as with MaxEnt, multiple knowledge sources can be integrated in a single model. While BoosTexter is known to be powerful when combining lexical and prosodic features within a single integral model, herein, I aim to evaluate how powerful it is when only a language model is used. In my experiments, the ICSI reimplementation of the original BoosTexter method was employed.[2] The used textual features had the same form as in the MaxEnt model.

2.4 Evaluation Metric

I measure DA segmentation performance using a "boundary error rate" (BER):

$$BER = \frac{I + M}{N} \quad [\%] \tag{3}$$

where I denotes the number of false DA boundary insertions, M the number of misses, and N the number of words in the test set.

3 Experimental Results

Table 1 presents experimental results for all three models (HMM, Maxent, and BoosTexter), all feature sets (words, AIC, POS, and POS mixed with words), and training and test conditions. The models for segmentation of human transcripts were trained on reference words. For testing on ASR data, I tried to use both true and recognized words for training, and compared performance of the models.

[1] `http://hal3.name/megam`
[2] `http://code.google.com/p/icsiboost/`

Table 1. DA segmentation results for individual language models, feature sets, and experimental setups [BER %] (REF=Reference human transcripts, ASR=Automatic transcripts, AIC=Automatically Induced Classes with 100 clusters, POS=Parts-of-speech, POSmixed=Parts-of-speech for infrequent words with 500 most frequent words kept. "Chance" refers to a model which classifies all test samples as within-DA boundaries.)

Model	**Used Features**	**Train/Test Setup**		
		REF/REF	**REF/ASR**	**ASR/ASR**
Chance	—	15.92%	13.85%	13.85%
HELM	Words	7.45%	9.41%	9.50%
	AIC	7.58%	9.70%	9.78%
	POS	10.62%	12.06%	11.85%
	POSmixed	7.65%	9.57%	9.59%
	Words+AIC	7.11%	9.25%	9.18%
	Words+POSmixed	7.23%	9.25%	9.31%
	Words+AIC+POSmixed	**7.02**%	**9.12**%	9.12%
MaxEnt	Words	7.50%	9.38%	9.38%
	AIC	7.42%	9.44%	9.37%
	POS	10.52%	11.79%	11.80%
	POSmixed	7.26%	9.23%	9.25%
	Words+AIC	7.19%	9.25%	9.21%
	Words+POSmixed	7.27%	9.27%	9.25%
	Words+AIC+POSmixed	**7.15**%	9.24%	**9.16**%
BoosTexter	Words	7.70%	9.52%	9.49%
	AIC	7.61%	9.50%	9.53%
	POS	10.87%	12.03%	11.13%
	POSmixed	7.68%	9.45%	9.46%
	Words+AIC	7.50%	9.42%	9.40%
	Words+POSmixed	7.66%	9.44%	9.45%
	Words+AIC+POSmixed	**7.46**%	**9.40**%	9.40%

In reference conditions, the best models based on a single feature group were MaxEnt for mixed POS and AICs, and HELM for words. On the other hand, the models only using POS information performed poorly. A comparison of POS and POSmixed shows that POS features are not sufficient indicators of DA boundaries and information provided by some frequent cue words is necessary to achieve satisfactory performance. In terms of a modeling approach comparison, it is interesting to observe that the generative HELM model is better in dealing with word information while the discriminative MaxEnt model better captures class information (both AIC and POSmixed). The BoosTexter model always performed worse than the other two models.

The results also indicate that an improvement is achieved when word information is combined with class information. The best result ($BER = 7.02\%$) is obtained when all three information sources are combined in the HELM model. The improvement over the baseline word-based model is statistically significant at $p < 10^{-23}$ using the Sign test. The difference between HELM and Boostexter is significant at $p < 10^{-13}$, and the difference between HELM and MaxEnt at $p < 0.02$. Of the other two models, MaxEnt outperformed BoosTexter (BER : 7.15% vs. 7.46%) which is significant at $p < 10^{-9}$.

As well as in reference conditions, MaxEnt for mixed POS was the best single model in ASR-based tests. Unlike reference conditions, MaxEnt was also the best model for capturing word information. The combination of all three knowledge sources was helpful once again, the best performing combined model was HELM ($BER = 9.12\%$) while BoosTexter was the worst. Both HELM and MaxEnt show a significant outperformance of the BoosTexter model ($p < 10^{-4}$). In contrast, the difference between HELM and MaxEnt is not significant.

A comparison of models trained on clean and erroneous data shows the following. While for HELM and BoosTexter the performance was almost the same, for the MaxEnt model, I got better results when training on automatic transcripts. However, even for the MaxEnt model, the difference in BER is only significant at $p < 0.08$.

4 Summary and Conclusions

I have explored the use of textual information for DA boundary detection in both human- and ASR-generated transcripts of multiparty meetings. I have analyzed contributions from word identities, parts-of-speech, and automatically induced word classes, and compared three statistical modeling approaches – HELM, MaxEnt, and BoosTexter.

Among others, the results indicate that POS information is only helpful when the most frequent words are kept and not replaced by POS tags. For both test conditions, the best results were achieved when all information sources were combined. The best performing combined model was HELM, achieving $BER = 7.02\%$ in reference and $BER = 9.12\%$ in ASR conditions. On the other hand, the boosting-based model was always the worst. While this model is powerful when combining prosodic and lexical information, it does not represent a good approach when only textual features are used and prosodic information is not accessible.

A comparison of models trained on clean and ASR data shows that for none of the models, significant improvement is achieved by training on ASR. The HELM and BoosTexter models perform approximately the same for both training setups, and the modest gain achieved by the ASR-trained MaxEnt model is not statistically significant.

Acknowledgments

This work was supported by the Ministry of Education of the Czech Republic under projects 2C06020 and ME909. In addition, I used the METACentrum computing clusters sponsored under the research program MSM6383917201. I also thank Yang Liu at UT Dallas for providing me with her models for the TnT tagger.

References

1. Jones, D., Wolf, F., Gibson, E., Williams, E., Fedorenko, E., Reynolds, D., Zissman, M.: Measuring the readability of automatic speech-to-text transcripts. In: Proc. Eurospeech, Geneva, Switzerland (2003)
2. Kahn, J.G., Ostendorf, M., Chelba, C.: Parsing conversational speech using enhanced segmentation. In: Proc. HLT-NAACL'04, Boston, MA, USA (2004)

3. Janin, A., Baron, D., Edwards, J., Ellis, D., Gelbart, D., Morgan, N., Peskin, B., Pfau, T., Shriberg, E., Stolcke, A., Wooters, C.: The ICSI meeting corpus. In: IEEE ICASSP 2003, Hong Kong (2003)
4. Dhillon, R., Bhagat, S., Carvey, H., Shriberg, E.: Meeting recorder project: Dialog act labeling guide. Technical Report TR-04-002, ICSI, Berkeley, CA, USA (2004)
5. Stolcke, A., Shriberg, E.: Automatic linguistic segmentation of conversational speech. In: Proc. ICSLP, Philadelphia, PA, USA (1996)
6. Warnke, V., Kompe, R., Niemann, H., Nöth, E.: Integrated dialog act segmentation and classification using prosodic features and language models. In: Proc. Europeech 97, Rhodes, Greece (1997)
7. Shriberg, E., Stolcke, A., Hakkani-Tür, D., Tür, G.: Prosody-based automatic segmentation of speech into sentences and topics. Speech Communication **32**(1-2) (2000) 127–154
8. Akita, Y., Saikou, M., Nanjo, H., Kawahara, T.: Sentence boundary detection of spontaneous Japanese using statistical language model and support vector machines. In: Proc. INTERSPEECH 2006 - ICSLP, Pittsburgh, PA, USA (2006)
9. Zimmermann, M., Hakkani-Tur, D., Fung, J., Mirghafori, N., Gottlieb, L., Shriberg, E., Liu, Y.: The ICSI+ multilingual sentence segmentation system. In: Proc. INTERSPEECH 2006 - ICSLP. (2006) 117–120
10. Kolář, J., Liu, Y., Shriberg, E.: Speaker adaptation of language models for automatic dialog act segmentation of meetings. In: Proc. INTERSPEECH 2007, Antwerp, Belgium (2007)
11. Stolcke, A., Chen, B., Franco, H., Gadde, V.R.R., Graciearena, M., Hwang, M.Y., Kirchhoff, K., Mandal, A., Morgan, N., Lei, X., Ng, T., Ostendorf, M., Sönmez, K., Venkataraman, A., Vergyri, D., Wang, W., Zheng, J., Zhu, Q.: Recent innovations in speech-to-text transcription at SRI-ICSI-UW. IEEE Trans. on Audio, Speech, and Language Processing **14**(5) (2006)
12. Liu, Y.: Structural Event Detection for Rich Transcription of Speech. PhD thesis, Purdue University, W. Lafayette, IN, USA (2004)
13. Brown, P., Pietra, V.D., de Souza, P., Lai, J., Mercer, R.: Class-based n-gram models of natural language. Computational Linguistics **18**(4) (1992) 467–479
14. Brants, T.: TnT – A statistical part-of-speech tagger. In: Proc. ANLP2000, Seattle, WA, USA (2000)
15. Chen, S., Goodman, J.: An empirical study of smoothing techniques for language modeling. Technical report, Harvard University, USA (1998)
16. Stolcke, A.: SRILM – An extensible language modeling toolkit. In: Proc. ICSLP'02, Denver, CO, USA (2002)
17. Berger, A., Della Pietra, S.A., Della Pietra, V.J.: A maximum entropy approach to natural language processing. Computational Linguistics **22**(1) (1996) 39–71
18. Schapire, R., Singer, Y.: BoosTexter: A boosting-based system for text categorization. Machine Learning **39**(2–3) (2000) 135–168
19. Kolář, J., Shriberg, E., Liu, Y.: Using prosody for automatic sentence segmentation of multiparty meetings. Lecture Notes in Artificial Intelligence (TSD'06) **4188** (2006) 629–636