Cross-Domain Speech Disfluency Detection

Kallirroi Georgila, Ning Wang, Jonathan Gratch
Institute for Creative Technologies, University of Southern California
12015 Waterfront Drive, Playa Vista, CA 90094, USA
{kgeorgila,nwang,gratch}@ict.usc.edu

Abstract

We build a model for speech disfluency detection based on conditional random fields (CRFs) using the Switchboard corpus. This model is then applied to a new domain without any adaptation. We show that a technique for detecting speech disfluencies based on Integer Linear Programming (ILP) (Georgila, 2009) significantly outperforms CRFs. In particular, in terms of F-score and NIST Error Rate the absolute improvement of ILP over CRFs exceeds 20% and 25% respectively. We conclude that ILP is an approach with great potential for speech disfluency detection when there is a lack or shortage of in-domain data for training.

1 Introduction

Speech disfluencies (also known as speech repairs) occur frequently in spontaneous speech and can pose difficulties to natural language processing (NLP) since most NLP tools (e.g. parsers and part-of-speech taggers) are traditionally trained on written language. However, speech disfluencies are *not* noise. They are an integral part of how humans speak, may provide valuable information about the speaker's cognitive state, and can be critical for successful turn-taking (Shriberg, 2005). Speech disfluencies have been the subject of much research in the field of spoken language processing, e.g. (Ginzburg et al., 2007).

Speech disfluencies can be divided into three intervals, the *reparandum*, the *editing term*, and the *correction* (Heeman and Allen, 1999; Liu et al., 2006). In the example below, "it left" is the reparandum (the part that will be repaired), "I mean" is the editing term, and "it came" is the correction:

```
(it left) * (I mean) it came
```

The asterisk marks the interruption point at which the speaker halts the original utterance in order to start the repair. The editing term is optional and consists of one or more filled pauses (e.g. uh, um) or discourse markers (e.g. you know, well). Our goal here is to automatically detect repetitions (the speaker repeats some part of the utterance), revisions (the speaker modifies the original utterance), or restarts (the speaker abandons an utterance and starts over). We also deal with complex disfluencies, i.e. a series of disfluencies in succession ("it it was it is sounds great").

In previous work many different approaches to detecting speech disfluencies have been proposed. Different types of features have been used, e.g. lexical features only, acoustic and prosodic features only, or a combination of both (Liu et al., 2006). Furthermore, a number of studies have been conducted on human transcriptions while other efforts have focused on detecting disfluencies from the speech recognition output.

In our previous work (Georgila, 2009), we proposed a novel two-stage technique for speech disfluency detection based on Integer Linear Programming (ILP). ILP has been applied successfully to several NLP problems, e.g. (Clarke and Lapata, 2008). In the first stage of our method, we trained state-of-the-art classifiers for speech disfluency detection, in particular, Hidden-Event Language Models (HELMs) (Stolcke and Shriberg, 1996), Maximum Entropy (ME) models (Ratnaparkhi, 1998), and Conditional Random Fields (CRFs) (Lafferty et al., 2001). Then in the second stage and during testing, each classifier proposed possible labels which were then assessed in the presence of local and global constraints using ILP. These constraints are hand-crafted and encode common disfluency patterns. ILP makes the

Proceedings of SIGDIAL 2010: the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pages 237–240,
The University of Tokyo, September 24-25, 2010. ©2010 Association for Computational Linguistics

final decision taking into account both the output of the classifier and the constraints. Our approach is similar to the work of (Germesin et al., 2008) in the sense that they also combine machine learning with hand-crafted rules. However, we use different machine learning techniques and ILP.

When we evaluated this approach on the Switchboard corpus (available from LDC and manually annotated with disfluencies) using lexical features, we found that ILP significantly improves the performance of HELMs and ME models with negligible cost in processing time. However, the improvement of ILP over CRFs was only marginal. These results were achieved when each classifier was trained on approx. 35,000 occurrences of disfluencies. Then we experimented with varying training set sizes in Switchboard. As soon as we started reducing the amount of data for training the classifiers, the improvement of ILP over CRFs rose and became very significant, approx. 4% absolute reduction of error rate with 25% of the training set (approx. 9,000 occurrences of disfluencies) (Georgila, 2009). This result showed that ILP is particularly helpful when there is no much training data available.

However, Switchboard is a unique corpus because the amount of disfluencies that it contains is very large. Thus even 25% of our training set contains more disfluencies than a typical corpus of human-human or human-machine interactions. In this paper, we investigate what happens when we move to a new domain when there is no in-domain data annotated with disfluencies to be used for training. This is usually the case when we start developing a dialogue system in a new domain, when the system has not been fully implemented yet, and thus no data from users interacting with the system has been collected. Since the improvement of ILP over HELMs and ME models was very large even when the models were both trained and tested on Switchboard (approx. 15% and 20% absolute reduction of error rate when 100% and 25% of the training set was used for training the classifiers respectively (Georgila, 2009)), in this paper we focus only on comparing CRFs versus CRFs+ILP. Our goal is to evaluate if and how much ILP improves CRFs in the case that no training data is available at all.

The structure of the paper is as follows: In section 2 we describe our data sets. In section 3 we concisely describe our approach. Then in section 4 we present our experiments. Finally in section 5 we present our conclusion.

2 Data Sets

To train our classifiers we use Switchboard (available from LDC), which is manually annotated with disfluencies, and is traditionally used for speech disfluency experiments. We transformed the Switchboard annotations into the following format:

```
it BE was IE a IP it was good
```

BE (beginning of edit) is the point where the reparandum starts and IP is the interruption point (the point before the repair starts). In the above example the beginning of the reparandum is the first occurrence of “it”, the interruption point appears after “a”, and every word between BE and IP is tagged as IE (inside edit). Sometimes BE and IP occur at the same point, e.g. “it BE-IP it was”. In (Georgila, 2009) we divided Switchboard into training, development, and test sets. Here we use the same training and development sets as in (Georgila, 2009) containing 34,387 occurrences of BE labels and 39,031 occurrences of IP labels, and 3,146 occurrences of BE labels and 3,499 occurrences of IP labels, respectively.

We test our approach on a smaller corpus collected in the framework of the Rapport project (Gratch et al., 2007). The goal of the Rapport project is to study how rapport is achieved in human-human and human-machine interaction. By rapport we mean the harmony, fluidity, synchrony and flow that someone feels when they are engaged in a good conversation.

The Rapport agent is a virtual human designed to elicit rapport from human participants within the confines of a dyadic narrative task (Gratch et al., 2007). In this setting, a speaker narrates some previously observed series of events, i.e. the events in a sexual harassment awareness and prevention video, and the events in a video of the Tweety cartoon. The central challenge for the Rapport agent is to provide the non-verbal listening feedback associated with rapportful interaction (e.g. head nods, postural mirroring, gaze shifts, etc.). Our ultimate goal is to investigate possible correlations between disfluencies and these types of feedback.

We manually annotated 70 sessions of the Rapport corpus with disfluencies using the labels described above (BE, IP, IE and BE-IP). In each session the speaker narrates the events of one video. These annotated sessions served as our reference data set (gold-standard), which contained 738 and 865 occurrences of BE and IP labels respectively.

3 Methodology

In the first stage we train our classifier. Any classifier can be used as long as it provides more than one possible answer (i.e. tag) for each word in the utterance. Valid tags are BE, BE-IP, IP, IE or O. The O tag indicates that the word is outside the disfluent part of the utterance. ILP will be applied to the output of the classifier during testing.

Let N be the number of words of each utterance and i the location of the word in the utterance (i=1,...,N). Also, let $C_{BE}(i)$ be a binary variable (1 or 0) for the BE tag. Its value will be determined by ILP. If it is 1 then the word will be tagged as BE. In the same way, we use $C_{BE-IP}(i)$, $C_{IP}(i)$, $C_{IE}(i)$, $C_O(i)$ for tags BE-IP, IP, IE and O respectively. Let $P_{BE}(i)$ be the probability given by the classifier that the word is tagged as BE. In the same way, let $P_{BE-IP}(i)$, $P_{IP}(i)$, $P_{IE}(i)$, $P_O(i)$ be the probabilities for tags BE-IP, IP, IE and O respectively. Given the above definitions, the ILP problem formulation can be as follows:

$$\begin{array}{l} max[\sum_{i=1}^{N}[P_{BE}(i)C_{BE}(i) + P_{BE-IP}(i)C_{BE-IP}(i) \\ +P_{IP}(i)C_{IP}(i) + P_{IE}(i)C_{IE}(i) + P_O(i)C_O(i)]] \end{array} \quad (1)$$

subject to constraints, e.g.:

$$\begin{array}{l} C_{BE}(i) + C_{BE-IP}(i) + C_{IP}(i) + C_{IE}(i) \\ +C_O(i) = 1 \qquad \forall i \in (1, ..., N) \end{array} \quad (2)$$

Equation 1 is the linear objective function that we want to maximize, i.e. the overall probability of the utterance. Equation 2 says that each word can have one tag only. In the same way, we can define constraints on which labels are allowed at the start and end of an utterance. There are also some constraints that define the transitions that are allowed between tags. For example, IP cannot follow an O directly, which means that we cannot start a disfluency with an IP. There has to be a BE after O and before IP. Details are given in (Georgila, 2009).

We also formulate some additional rules that encode common disfluency patterns. The idea here is to generalize from these patterns. Below is an example of a long-context rule. If we have the sequence of words "she was trying to well um she was talking to a coworker", we expect this to be tagged as "she BE was IE trying IE to IP well O um O she O was O talking O to O a O coworker O", if we do not take into account the context in which this pattern occurs. Basically the pattern here is that two sequences of four words separated by a discourse marker ("well") and a filled pause ("um") differ only in their third word. That is, "trying" and "talking" are different words but have the same part-of-speech tag (gerund). We incorporate this rule into our ILP problem formulation as follows: Let (w_1,...,w_N) be a sequence of N words where both w_3 and w_{N-3} are verbs (gerund), the word sequence w_1,w_2,w_4 is the same as the sequence w_{N-5},w_{N-4},w_{N-2}, and all the words in between (w_5,...,w_{N-6}) are filled pauses or discourse markers. Then the probabilities given by the classifier are modified as follows: $P_{BE}(1)$=$P_{BE}(1)$+$b1$, $P_{IE}(2)$=$P_{IE}(2)$+$b2$, $P_{IE}(3)$=$P_{IE}(3)$+$b3$, and $P_{IP}(4)$=$P_{IP}(4)$+$b4$, where $b1$, $b2$, $b3$ and $b4$ are empirically set boosting paremeters with values between 0.5 and 1 computed using our Switchboard development set. We use more complex rules to cover cases such as "she makes he doesn't make", and boost the probabilities that this is tagged as "she BE makes IP he O doesn't O make O".

In total we apply 17 rules and each rule can have up to 5 more specific sub-rules. The largest context that we take into account is 10 words, not including filled pauses and discourse markers.

4 Experiments

For building the CRF model we use the CRF++ toolkit (available from `sourceforge`). We used only lexical features, i.e. words and part-of-speech (POS) tags. Switchboard includes POS information but to annotate the Rapport corpus with POS labels we used the Stanford POS tagger (Toutanova and Manning, 2000). We experimented with different sets of features and we achieved the best results with the following setup (i is the location of the word or POS in the sentence): Our word features are $\langle w_i \rangle$, $\langle w_{i+1} \rangle$, $\langle w_{i-1}, w_i \rangle$, $\langle w_i, w_{i+1} \rangle$, $\langle w_{i-2}, w_{i-1}, w_i \rangle$, $\langle w_i, w_{i+1}, w_{i+2} \rangle$. Our POS features have the same structure as the word features. For ILP we use the `lp_solve` software also available from `sourceforge`. We train on Switchboard and test on the Rapport corpus.

For evaluating the performance of our models we use standard metrics proposed in the literature, i.e. Precision, Recall, F-score, and NIST Error Rate. We report results for BE and IP. F-score is the harmonic mean of Precision and Recall (we equally weight Precision and Recall). Precision is the ratio of the correctly identified tags X to all the tags X detected by the model (where X is BE or IP). Recall is the ratio of the correctly identified tags X to all the tags X that appear in the reference

	BE			
	Prec	**Rec**	**F-score**	**Error**
CRF	74.52	36.45	48.95	73.44
CRF+ILP	77.44	64.63	70.46	47.56
	IP			
	Prec	**Rec**	**F-score**	**Error**
CRF	86.36	41.73	56.27	64.62
CRF+ILP	88.75	72.95	80.08	35.61

Table 1: Comparative results between our models.

utterance. The NIST Error Rate is the sum of insertions, deletions and substitutions divided by the total number of reference tags (Liu et al., 2006).

Table 1 presents comparative results between our models. As we can see, now the improvement of ILP over CRFs is not marginal as in Switchboard. In fact, in terms of F-score and NIST Error Rate the absolute improvement of ILP over CRFs exceeds 20% and 25% respectively. The results are statistically significant ($p<10^{-8}$, Wilcoxon signed-rank test). The main gain of ILP comes from the large improvement in Recall. This result shows that using ILP has great potential for speech disfluency detection when there is a lack of in-domain data for training, and when we use lexical features and human transcriptions. Furthermore, the cost of applying ILP is negligible since the process is fast and applied during testing.

Note that the improvement of ILP over CRFs is significant even though the two corpora, Switchboard and Rapport, differ in genre (conversation versus narrative).

The reason for the large improvement of ILP over CRFs is the fact that as explained above ILP takes into account common disfluency patterns and generalizes from them. CRFs can potentially learn similar patterns from the data but do not generalize that well. For example, if the CRF model learns that “she she” is a repetition it will not necessarily infer that any sequence of the same two words is a repetition (e.g. “and and”). Of course here, since we deal with human transcriptions we do not worry about speech recognition errors. Preliminary results with speech recognition output showed that ILP retains its advantages but more modestly. In this case, when deciding which boosting rules to apply, it makes sense to consider speech recognition confidence scores per word. For example, a possible repetition “to to” could be the result of a misrecognition of “to do”. But these types of problems also affect plain CRFs, so in the end ILP is expected to continue outperforming CRFs. This is one of the issues for future work together with using prosodic features.

5 Conclusion

We built a model for speech disfluency detection based on CRFs using the Switchboard corpus. This model was then applied to a new domain without any adaptation. We showed that a technique for detecting speech disfluencies based on ILP significantly outperforms CRFs. In particular, in terms of F-score and NIST Error Rate the absolute improvement of ILP over CRFs exceeds 20% and 25% respectively. We conclude that ILP is an approach with great potential for speech disfluency detection when there is a lack or shortage of in-domain data for training.

Acknowledgments

This work was sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM). The content does not necessarily reflect the position or the policy of the Government, and no official endorsement should be inferred.

References

J. Clarke and M. Lapata. 2008. Global inference for sentence compression: An integer linear programming approach. *Journal of Artificial Intelligence Research*, 31:399–429.

K. Georgila. 2009. Using integer linear programming for detecting speech disfluencies. In *Proc. of NAACL.*

S. Germesin, T. Becker, and P. Poller. 2008. Hybrid multi-step disfluency detection. In *Proc. of MLMI.*

J. Ginzburg, R. Fernández, and D. Schlangen. 2007. Unifying self- and other-repair. In *Proc. of DECALOG.*

J. Gratch, N. Wang, J. Gerten, E. Fast, and R. Duffy. 2007. Creating rapport with virtual agents. In *Proc. of International Conference on Intelligent Virtual Agents (IVA).*

P. Heeman and J. Allen. 1999. Speech repairs, intonational phrases and discourse markers: Modeling speakers' utterances in spoken dialogue. *Computational Linguistics*, 25:527–571.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. of ICML.*

Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, and M. Harper. 2006. Enriching speech recognition with automatic detection of sentence boundaries and disfluencies. *IEEE Trans. Audio, Speech and Language Processing*, 14(5):1526–1540.

A. Ratnaparkhi. 1998. *Maximum Entropy Models for natural language ambiguity resolution.* Ph.D. thesis, University of Pennsylvania.

E. Shriberg. 2005. Spontaneous speech: How people really talk, and why engineers should care. In *Proc. of Interspeech.*

A. Stolcke and E. Shriberg. 1996. Statistical language modeling for speech disfluencies. In *Proc. of ICASSP.*

K. Toutanova and C.D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proc. of EMNLP/VLC.*