

Character Identification on Multiparty Dialogue based on End-to-End Neural Coreference Resolution

Pu-Chin Chen {puchinchen@ucla.edu}
Aoxuan Li {a811278305@gmail.com}
Yutian Zhang {yutianzh0527@gmail.com}
Xin Liu {xinliu627@ucla.edu}

Abstract

This document contains the instructions for preparing a camera-ready manuscript for the proceedings of ACL-2016. The document itself conforms to its own specifications, and is therefore an example of what your manuscript should look like. These instructions should be used for both papers submitted for review and for final versions of accepted papers. Authors are asked to conform to all the directions reported in this document.

1 Introduction

Our main goal is to accomplish a shared task in SemEval 2018 - Task 4: Character Identification on Multiparty Dialogues. This task requires us to build a system which can identify different mentions in multiparty dialogues as corresponding characters in the show. This task is rather challenging, for cross-document entity resolution is imperative for identifying such mentions as real characters.

Literally, the character identification problem is tackled as a coreference resolution task with a further step on entity linking. In terms of this task, the baseline model generates mentions from a coreference system, and then each coreference chain is linked to a specific character identity. Both parts are implemented with agglomerative convolutional neural network in previous system (Chen et al., 2017; Chen and Choi, 2016). Alternatively, we intend to use Bidirectional-LSTM with attention mechanism to address the same problem, which proves to have a satisfying performance in many NLP tasks. In our project, we apply the end-to-end neural coreference resolution model introduced by Lee and He, etc. (Lee et al., 2017). After obtaining the predicted clusters of mentions by

the end-to-end coreference system, we choose the same algorithm of entity-linking as the one Chen proposed in his paper.

2 Previous Work

2.1 Coreference resolution

Machine learning methods have a long history in coreference resolution. However, the learning problem is challenging and, until very recently, hand-engineered systems built on top of automatically produced parse trees (Raghunathan et al., 2010) outperformed all learning approaches. Durrett and Klein (2013) showed that highly lexical learning approaches reverse this trend, and more recent neural models (Wiseman et al., 2016; Clark and Manning, 2016b,a) have achieved significant performance gains. However, all of these models still use parsers for head features and include highly engineered mention proposal algorithms. Such pipelined systems suffer from two major drawbacks: (1) parsing mistakes can introduce cascading errors and (2) many of the hand-engineered rules do not generalize to new languages or domains.

The end-to-end coreference resolution model we used in this task significantly outperforms all previous work without using a syntactic parser or hand-engineered mention detector.

2.2 Entity Linking

Entity linking has traditionally relied heavily on knowledge databases, most notably, Wikipedia, for entities (Mihalcea and Csomai, 2007b; Ratnikov et al., 2011b; Gattani et al., 2013; Francis-Landau et al., 2016). Although we do not make use of knowledge bases, our task is closely aligned to entity linking. Recent advances in entity linking are also applicable to our task since we see Francis-Landau et al. (2016) use convolutional

nets to capture semantic similarity between a mention and an entity by comparing context of the mention with the description of the entity. This work validates our usage of deep learning for character identification.

Dialogue tracking has been an expanding task as shown by the Dialogue State Tracking Challenges hosted by Microsoft (Kim et al., 2015). That an ongoing conversation can be dynamically tracked (Henderson et al., 2013) is exciting and applicable to our task because the state of a conversation may yield significant hints for entity linking and coreference resolution. Speaker identification, a task similar to character identification, has already shown some success with partial dialogue tracking by dynamically identifying speakers at each turn in a dialogue using conditional random field models.

3 End-to-End Coreference Resolution

4 Entity Linking

5 Dataset and Evaluation Metrics

5.1 Data Description

The data comes from the scripts of the first two seasons of TV show Friends, which are already annotated for this task. Specifically, each season consists of episodes, and each episode is comprised of scenes. Furthermore, each scene is segmented into sentences. All datasets follow the CoNLL 2012 Shared Task data format. Each sentence is delimited by a new line and each column indicates the information regarding the word or the punctuation.

5.2 Data Split

Considering the amount of total data is not so sufficient, we simply split them into two datasets for different purposes, training and evaluation sets. The exact ratio of our training set to our evaluation set is 4:1. The evaluation set contains seventy-four scenes in *Friends*.

5.3 Coreference Evaluation Metrics

The coreference results are evaluated with three metrics apropos coreference resolution: MUC, B^3 , and $CEAF_e$. The precision, recall and F1 score of each metric approach are calculate separately and then the averages of them are obtained, which are utilized to compare against each other.

MUC

MUC(Vilain et al., 1995) concerns the number of pairwise links needed to be inserted or removed to map system responses to gold keys. The number of links shared by system and gold are calculated. In addition, minimum numbers of links needed to describe coreference chains of the system and gold are computed as well.

B^3

Rather than evaluating coreference chains merely on their links, B^3 (Bagga and Baldwin, 1998) metric computes precision and recall on a mention level. System performance is evaluated by the average of all mentions scores.

$CEAF_e$

$CEAF_e$ (Luo, 2005) metric further clarifies the downside of B^3 , in which entites can be used more than once during evaluation. Consequently, both multiple coreference chains of the same entity and chains with mentions of multiple entities are not penalized. To mitigate the aforementioned problem, $CEAF_e$ evaluates exclusively on the best one-to-one mapping between the systems and golds entities.

6 Results and Analysis

We conducted one complete experiment and two ablation experiments, no heads and no features. The result of no features is shown in Table 1.

7 Conclusion and Future work

Acknowledgments

The acknowledgments should go immediately before the references. Do not number the acknowledgments section. Do not include this section when submitting your paper for review.

A Supplemental Material

ACL 2016 also encourages the submission of supplementary material to report preprocessing decisions, model parameters, and other details necessary for the replication of the experiments reported in the paper. Seemingly small preprocessing decisions can sometimes make a large difference in performance, so it is crucial to record such decisions to precisely characterize state-of-the-art methods.

Nonetheless, supplementary material should be supplementary (rather than central) to the paper. It may include explanations or details of proofs or

200	deriations that do not fit into the paper, lists of fea-	250
201	tures or feature tempates, sample inputs and out-	251
202	puts for a system, pseudo-code or source code, and	252
203	data. (Source code and data should be separate up-	253
204	loads, rather than part of the paper).	254
205	The paper should not rely on the supplementary	255
206	material: while the paper may refer to and cite the	256
207	supplementary material will be available to the re-	257
208	viewers, they will not be asked to review the sup-	258
209	plementary material.	259
210	Appendices (i.e. supplementary material in the	260
211	form of proofs, tables, or pseudo-code) should	261
212	come after the references, as shown here. Use	262
213	<code>\appendix</code> before any appendix section to	263
214	switch the section numbering over to letters.	264
215	B Multiple Appendices	265
216	... can be gotten by using more than one section.	266
217	We hope you won't need that.	267
218		268
219		269
220		270
221		271
222		272
223		273
224		274
225		275
226		276
227		277
228		278
229		279
230		280
231		281
232		282
233		283
234		284
235		285
236		286
237		287
238		288
239		289
240		290
241		291
242		292
243		293
244		294
245		295
246		296
247		297
248		298
249		299