HW 4

Deliverable 7.1 (6 points):

Describe your bakeoff design. What worked and what didn't? Give a possible reason behind it.

I used the pretrained embedding and used the same architecture as designed in 4.5. I added two more layers to the AttentionBasedMarkableEmbedding module. I fine-tuned the learning rate. I found that SDG optimizer along with a learning rate of 0.002 worked well on the data. I combined the fairy tale and wsj to train the network and achieved a F1 score of 60 on the dev set.

I performed the a lot of parameter tuning, the following experiments did not work:

Experiments that did not work:

i) I used only all_words and all_markables for training and did not include the fairy dataset. Using these as the dataset, I performed the following parameter tuning:

a) Used learning rate = 0.001, epoch = 25, optimizer = Adam -> Dev set F score = 56.36
b) Used learning rate = 0.005, epoch = 25, optimizer = SDG -> Dev set F score = 55.58
c) Added one more layer in Attention Module, and then repeated a) and b) and received the Dev set F scores as 54.4 and 53 respectively.

I also tried running all the above experiments for 50 epochs but the results deteriorated. Possible reasons for not working - I think my model was overfitting, as the longer I trained, the F score became smaller. Also, during the training, for part b) and c), I observed that the training loss was not constantly decreasing, instead it was fluctuating at times. The training did not seem very stable.

Experiments that worked:

- In this experiment, I first combined the all_words, all_markables with the fairy tale data. This resulted in larger training data set and improved the performance of the resolver:
- I performed the following experiments:
  a) Used learning rate = 0.003, optimizer = SDG, epochs = 50-> F1 on Dev set = 59.02

b) Used learning rate = 0.005, optimizer = SDG, epochs = 50-> F1 on Dev set = 59.41
c) Added one more layer to the Attention Module and used learning rate = 0.003, optimizer = SDG, epochs = 50-> F1 on Dev set = 60.03

These experiments worked and led to the final result.

Deliverable 7.2 (8 points):

You will select a research paper at ACL, EMNLP or NAACL that focuses on coreference resolution. Summarize the paper, answering the following questions:

Paper - Lexical Features in Coreference Resolution: To be Used With Caution

Authors - Nafise Sadat Moosavi, Michael Strube

Venue - Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics

1. **What are the main ideas of the paper?**

In this paper, the authors investigate a drawback of using many lexical features in state-of-the-art coreference resolvers. They show that if coreference resolvers mainly rely on lexical features, they can hardly generalize to unseen domains. Authors present a comparative study between coreference resolvers using lexical features and those which did not use lexical features. Authors also argue that the current coreference resolution evaluation is flawed as it evaluates on a specific split of a specific dataset in which there is a notable overlap between the training, development, and test sets.

2. **What worked and what didn't?**

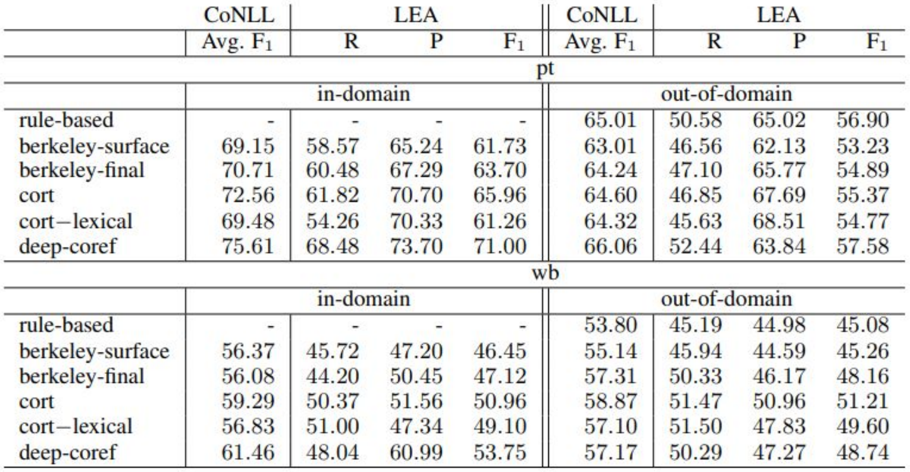
Coreference resolution is an important step for text understanding and usually, it is not an end task. Coreference resolvers will be used in tasks and domains for which coreference annotated corpora may not be available. Therefore, generalizability should be considered as an important parameter while developing coreference resolvers.

Authors show that if lexical features are used for coreference resolution, then resolvers do not generalize well on unseen domains. Authors present proof that state-of-the-art coreference resolvers trained on the CoNLL dataset perform worse than the rule-based system (Lee et al., 2013), on the new dataset, WikiCoref (Ghaddar and Langlais, 2016b), even though WikiCoref is annotated with the same annotation guidelines as the CoNLL dataset. The authors also presented a comparison of coreference resolvers on this dataset to prove this.

Authors show that extensive use of lexical features biases coreference resolvers towards seen mentions. Authors show that there is a notable overlap between training, test, and development datasets in CoNLL that encourages overfitting. Authors encourage incorporating out-of-domain evaluations in the current coreference evaluation scheme. Out-of-domain evaluations could be performed by using either the existing genres of the CoNLL dataset or by using other existing coreference annotated datasets like WikiCoref, MUC, or ACE.

3. **Are there any rationales the paper provided or you think that lead to the results?**

Authors present claims and then present discussions or empirical results to support their claim. Authors claim that if coreference resolvers mainly rely on lexical features, they can don't generalize well to unseen domains. Authors perform a comparison between in-domain and out-of-domain evaluations of the resolvers that used lexical features. The below table shows the results. Authors point that significance drop in out-of-domain performance is due to overfitting and it indicates that the model does not generalize well.



	CoNLL	LEA			CoNLL	LEA		
	Avg. F_1	R	P	F_1	Avg. F_1	R	P	F_1
pt								
	in-domain				out-of-domain			
rule-based	-	-	-	-	65.01	50.58	65.02	56.90
berkeley-surface	69.15	58.57	65.24	61.73	63.01	46.56	62.13	53.23
berkeley-final	70.71	60.48	67.29	63.70	64.24	47.10	65.77	54.89
cort	72.56	61.82	70.70	65.96	64.60	46.85	67.69	55.37
cort−lexical	69.48	54.26	70.33	61.26	64.32	45.63	68.51	54.77
deep-coref	75.61	68.48	73.70	71.00	66.06	52.44	63.84	57.58
wb								
	in-domain				out-of-domain			
rule-based	-	-	-	-	53.80	45.19	44.98	45.08
berkeley-surface	56.37	45.72	47.20	46.45	55.14	45.94	44.59	45.26
berkeley-final	56.08	44.20	50.45	47.12	57.31	50.33	46.17	48.16
cort	59.29	50.37	51.56	50.96	58.87	51.47	50.96	51.21
cort−lexical	56.83	51.00	47.34	49.10	57.10	51.50	47.83	49.60
deep-coref	61.46	48.04	60.99	53.75	57.17	50.29	47.27	48.74

Table 3: In-domain and out-of-domain evaluations for a high and a low overlapped genres.

Authors show that there is a notable overlap between training, test, and development datasets in CoNLL and claim that this may be the possible reason for overfitting. Authors present empirical figures for overlap.

So, authors encourage incorporating out-of-domain evaluations in the current coreference evaluation scheme to avoid overfitting. Hence, authors conclude that lexical features should be used with caution in coreference resolvers.

4. **How would you further improve over this work if you have the opportunity?**

   If given a chance to improve the paper, I would include/perform the following:

   1) Authors suggest that out-of-domain evaluations should be incorporated to avoid overfitting. However, the authors do not present sufficient empirical evidence and discussion for this part. I believe that a small study with empirical results showing the effect of including out-of-domain evaluations will further support the idea presented by the authors.
   2) Authors also mention that pruning rare lexical features plus incorporating more generalizable features could also help to prevent overfitting. However, there is no discussion on what kind of 'general features' should be included. Also, there is no information about how pruning should be performed, how should the threshold for pruning be decided etc.

      Overall, I feel that authors have focused the paper on highlighting the drawbacks and less on the solution to the problem. I think that if given a chance to improve the paper, I would also like to include more details and experiments on the solution to the overfitting part.

Deliverable 7.3 (4 points):

Pick any topic that you like or learned a lot from this course. Propose a related quiz question and give an answer to it. Please feel free to be creative here and remember that we don't have final exams for this iteration of the course :)

Can we use one-hot encoding to represent a word? What are the possible problems of this approach?

- Yes, one-hot encoding can be used. The length of vector will be same as the size of the vocabulary. All values will be zero except for the index corresponding to the given word.
- Problems - If the size of the vocabulary is large, so the vector will be a large sparse vector; also in this representation, we do not capture the similarity between similar words.

Which two conflicting constraints are captured by tf-idf?

- Term frequency(tf) - frequency of word t in the document
- Inverse document frequency(idf) - it is the log of the ratio of the number of documents in the corpus to the number of documents that a term occurs in.

What are the disadvantages of using n-grams? Give possible solutions.

- N-grams are computationally expensive. It is not possible to compute n-grams for n>5. Also, for small n, it can't keep track of long-distance context.
- Solutions - i) pruning - keep only those n-grams with count>threshold
  ii) improve computational efficiency by using efficient data structures like trees, perform quantization to store probabilities.