XiaoshuoNLP: An NLP Pipeline for Processing Chinese Literary Texts

Kiara Meng Hui Liu*, Xiaomeng Zhu*, Carolyn Jane Anderson
Department of Computer Science, Wellesley College

Background

There currently exist multiple NLP tools and pipelines for processing English literary texts; they are able to perform tasks such as tokenization, part-of-speech tagging, named entity recognition, and coreference resolution (Bamman et al., 2014; Yoder et al., 2021). Specifically designed for processing fiction, these NLP systems allow researchers and scholars alike to efficiently extract information from literary narratives. However, even though Chinese is one of the most high-resource languages in the world, there is no similar pipeline in Chinese yet, potentially due to its relative lack of annotated data in specifically the fiction genre.

Corpus

From a linguistic perspective, Chinese literature can be separated into two categories: the vernacular style and the literary (classical) style.

Classical/Literary (文言文):

- Historically used for written documents and literary text
- Shorter average token length
- Does not reflect spoken language and was inaccessible to most people

Vernacular (白话文):

- Came into popularity during the New Culture Movement (1910-1920) and has been widely used ever since
- Longer average token length
- Better reflects spoken language and is accessible to more people

Since there is often incorporation of one style within another, it is helpful to think of the style variation as a spectrum: on one side is literary Chinese, which is rather difficult for current-day readers to understand; on the other side is vernacular Chinese, which is used extensively by contemporary writers. In the middle of the spectrum are novels written between the Song Dynasty and the Qing Dynasty, which are primarily in the vernacular style but contain some characteristics of simple classical Chinese.

We have gathered 10 texts from Project Gutenberg. Genres include Ming-Qing novels, which involve a fusion of vernacular and classical Chinese (*Dream of the Red Chamber, Journey to the West, The Plum in the Golden Vase, The Fox and the Fate, A Flower in a Sinful Sea*), modern short stories (*The True Story of Ah-Q, Diary of a Madman, Abundant Harvest*), *sanwen* (散文) prose (*Dawn Blossoms Plucked at Dusk*), and the literary diary (*Ling li ji guang*). We used the first one or two chapters of each text to create annotated data for evaluating model performance and later training our own cluster merging model.

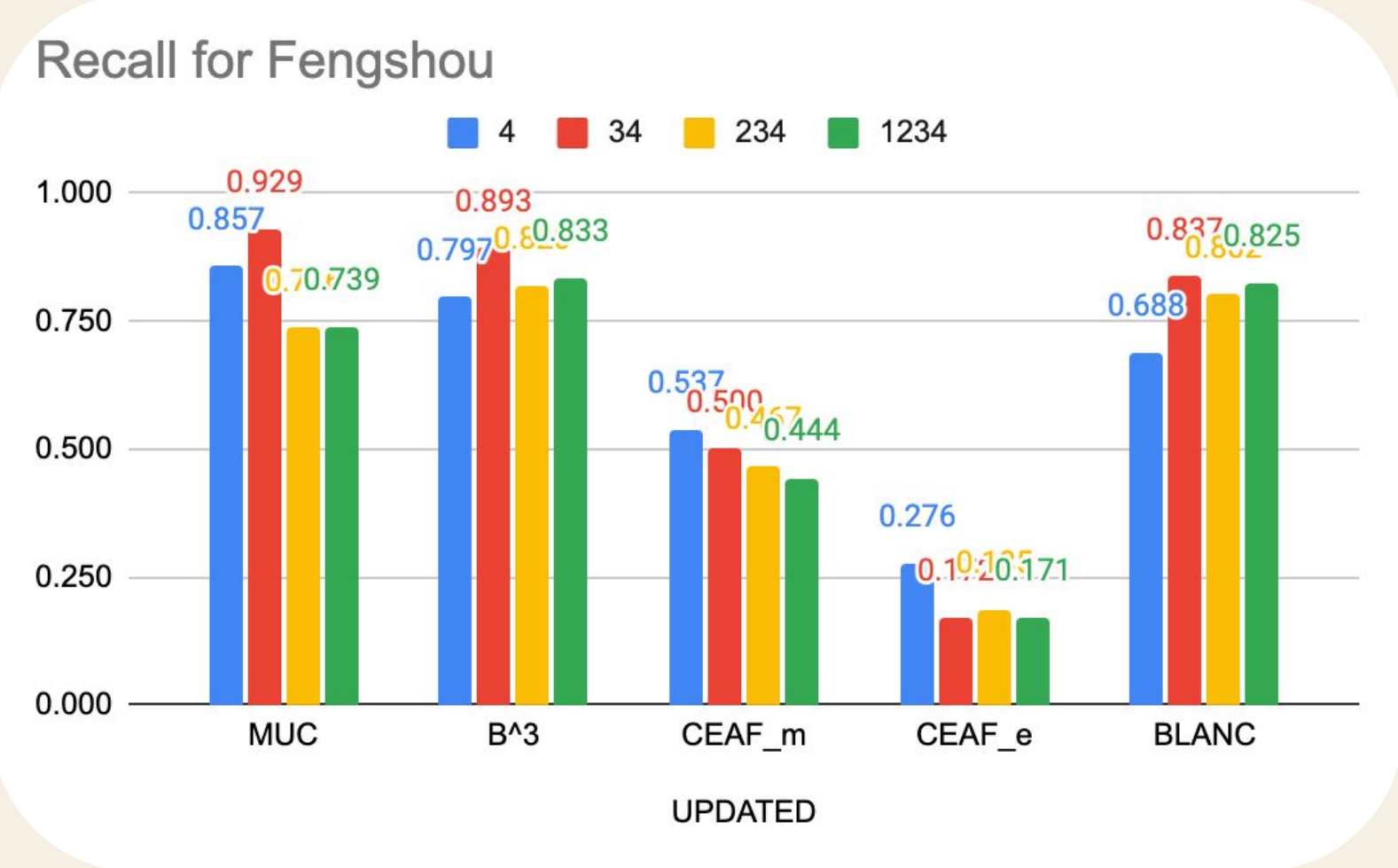

Figure 1

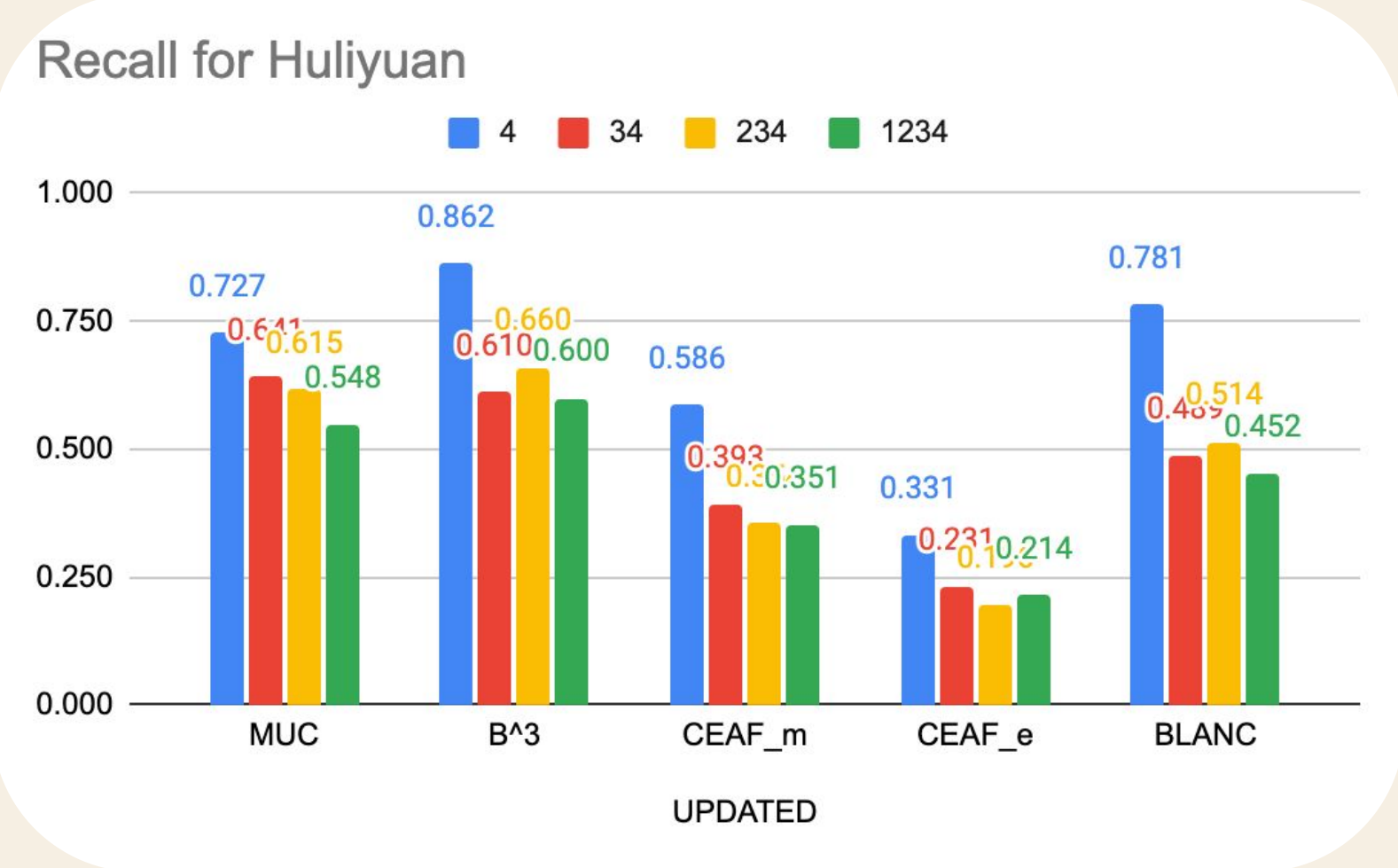

Figure 2

*Equal contributions

Tasks & Evaluation

Tokenization:

- Tokenization is the task of splitting text into "tokens," or basic linguistic units. In English, words are separated by spaces, but this is not the case in Chinese, which makes the task of tokenization particularly difficult.
- We evaluated 6 models and found that **HanLP** resulted in the highest performance when compared to our annotations using the minimum edit distance algorithm.

Part-of-Speech Tagging (POS):

- Part-of-Speech Tagging is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech based on both its definition and its context.
- Some of the existing models that support part-of-speech tagging were not built to take in customized tokens. To account for differences in tokens, we calculated the minimum edit distance for two lists of (token, part-of-speech) pairs. For models that support customized tokens, we calculated the average percentage of mismatch per sentence.
- Comparison against our gold standard annotations indicates that **HanLP** performs best in both metrics.

Named Entity Recognition (NER):

- Named entity recognition is the task of extracting named entities (such as characters, locations, organizations, etc.) from the text. This is also more difficult in Chinese than in English, since Chinese has no capitalization of names.
- Most models only have NER tags incorporated into their POS components, but **HanLP** has a separate model dedicated to NER.
- HanLP's tagging conventions are different from ours: when honorifics are present, our annotations include the entire name-honorific pair as a named entity, but HanLP includes only the name. To account for this difference, we compiled a list of 200+ common Chinese honorifics for comparison across the two different conventions and appended the honorifics to our final named entities output.

Coreference Resolution:

- Coreference resolution is the task of finding all expressions that refer to the same entity in a text. This can be difficult in some Chinese texts, especially older ones, since characters often have several names (surname, given name, nickname, and title, to *name* a few) and can be referred to as various combinations of the above.
- HanLP's coreference resolution component recognizes not only character coreferences, but also other entities such as locations and time. We filter these results for characters by using both the previous NER outputs and a set of character-related keywords (such as pronouns, honorifics, etc.).
- We investigated whether providing the model with longer spans of context can improve its performance on coreference resolution. Results (Figure 1 & Figure 2) show that as we feed more sections of text into the model, the recall value decreases, suggesting that in order for the pipeline to achieve the best performance, we need to segment the text into appropriate lengths and merge clusters across sections.

	jieba	LAC	HanLP👑	jiagu	THULAC	pkuseg
Tokenization	824	675	563	628	641	626
Part-of-Speech Tagging	0.59	0.35	0.24	0.31	0.30	0.31
	/	/	0.13	0.19	/	/
Named Entity Recognition	/	0.71	0.79	0.16	0.48	/
Coreference Resolution	/	/	See Figure 1 & 2	/	/	/

Table 1

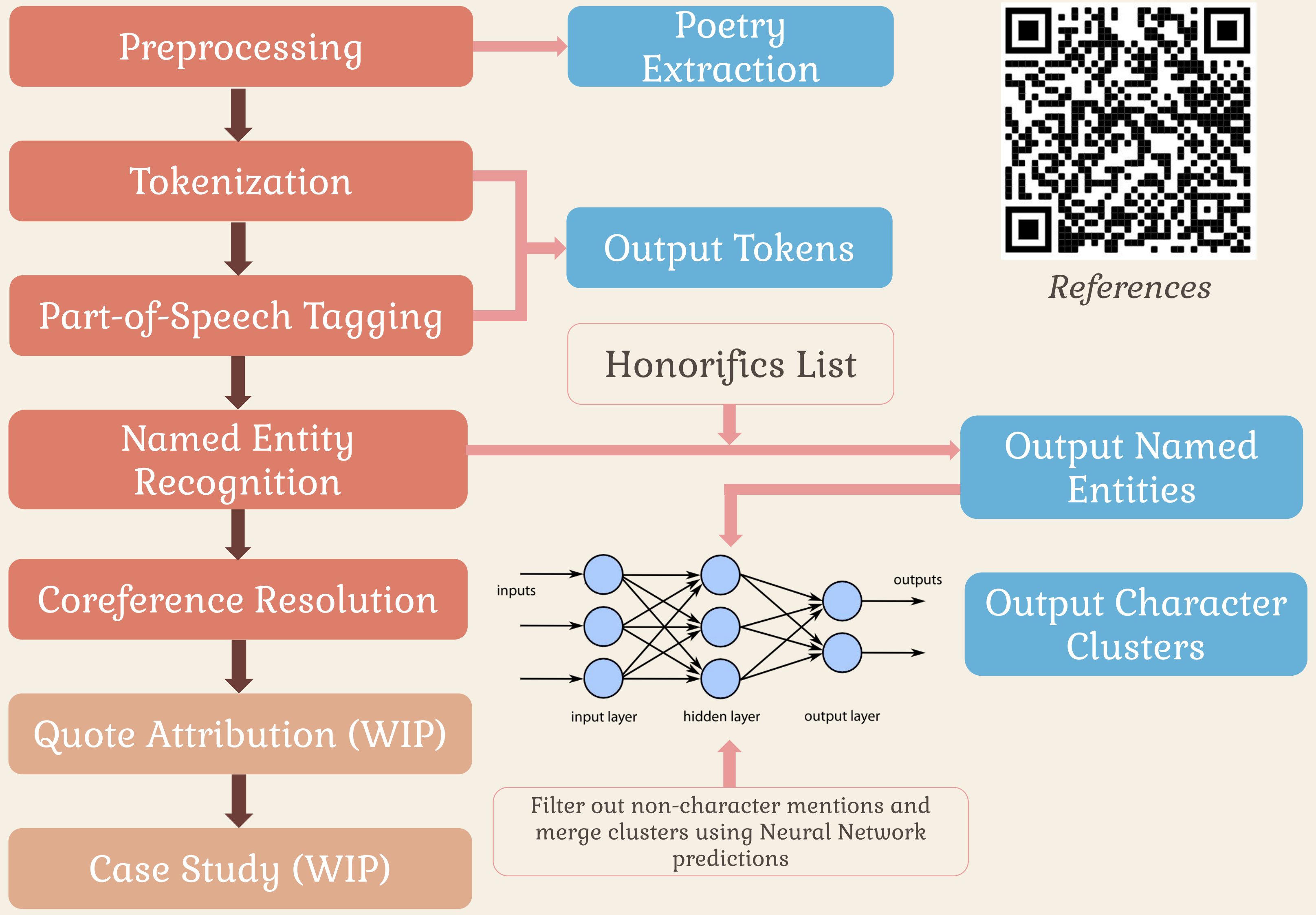

Figure 3

References

Cluster Merging

Dataset: We picked 4 excerpts from 4 texts of various styles from our corpus and annotated them on character clusters. Cluster pairs are constructed by running the HanLP coreference resolution tool on sections of around 1000 characters (which was discovered in previous experiments to be the length of texts that HanLP performs best on). For all clusters produced by HanLP, we enumerated all combinations of 2 and consider each one as a datapoint in the dataset.

Features: For each cluster pair, we calculated the following 5 features:

- RoBERTa embedding of the most frequent mention in the first cluster
- RoBERTa embedding of the most frequent mention in the second cluster
- Minimum edit distance between the most frequent mentions in each cluster
- Size difference between the two clusters
- The minimum difference in index between mentions in both clusters

Labels: Each mention in the cluster was assigned a cluster ID according to our gold standard annotations. We assigned 1 (merge) to the cluster pair if cluster purity increases after merging two clusters in the pair together. 0 (no merge) was assigned otherwise.

Training: We are training a neural network that takes a cluster pair as input and predicts if the two clusters in the pair should be merged.

Future Work

Quote Attribution: This will be a component that identifies the speaker of every quote. There are two steps to this task: the first is quotation identification, and the second is speaker attribution. The former can be achieved through searching for quotation marks via simple regular expressions, while the latter would likely require us to train our own model.

Case Study: We plan to evaluate the performance of our entire pipeline by conducting a case study on some previously unseen text. We also hope to conduct a case study on a translated work, comparing the performance of our pipeline on the Chinese version to the performance of a similar English pipeline on the English version.