ANLP Assignment 3 2014

due: Monday 24 November, 3pm, electronic submission (see end of instructions)

1 Overview

This coursework deals with the problem of cross-document co-reference resolution (CDCR), i.e. identifying which referring expressions (mentions) from different documents refer to the same entity.

Just like within-document co-reference resolution systems, CDCR systems need to cope with two forms of ambiguity. First, polysemy, where the same name (like John Smith) could refer to different entities. Second, synonymy, where the same entity could be referred to using different names. This could be due to spelling errors, alternative referring expressions (John Smith, Mr. Smith, or Smith), name changes (fairly common for celebrities, or after marriage), or title changes (when someone becomes a doctor or ascends to a throne). Successful CDCR systems need to cope with both forms of ambiguity. This is achieved by integrating information about the context in which the entity is mentioned along with the mention text itself.

We have already implemented a complete CDCR system for you. The system currently predicts which mentions refer to the same entity using some simple measures of similarity between mentions, which we describe in more detail below. Your task in this assignment is to explore other ways of determining similarity, considering in particular what features might be important in measuring similarity between mentions. Your work here is intended to go a little beyond just what is covered in the textbook and lectures; we have provided a few references to relevant papers that might give you some ideas, or you might wish to do some research of your own.

Your mark on this assignment will be determined by how well you are able to justify the choice of features and similarity measures you choose to explore, as well as the presentation, analysis, and discussion of your experimental results. For more details, see the marking guidelines below.

2 Working with others (and not)

You may complete this assignment with a partner. You are free to discuss any aspects of the assignment with your partner and divide up the tasks however you wish; however we would encourage you to collaborate on each part rather than doing a strict division of tasks, as this will enable better learning for both of you.

You are also free to discuss high-level concepts and general programming questions with others in the class; however you may NOT share code or answers directly with other groups. For example, it is ok to suggest using a particular library function to help solve a task; it is not ok to send (or describe) a code snippet showing exactly how you used that function on what arguments. Your code and report must be your own group’s work.

Please see the School’s guidance on academic misconduct for further information:

---

1The cross-document mentions you will be working with here do not include pronominal mentions.
3 Further details

3.1 System overview

Our system implements CDCR as a classification task: for each pair of mentions, the system predicts whether those mentions refer to the same entity or not.\footnote{Note that many other systems treat CDCR as a \textit{clustering} task, attempting to group together all mentions of the same entity into a single cluster. Within a cluster, all pairs of mentions would be predicted to refer to the same entity. Since our system classifies mention pairs independently of each other, it is possible that for three mentions \(A, B, C\), the system might predict that \(A\) and \(B\) are co-referent, \(B\), and \(C\) are co-referent, but \(A\) and \(C\) are not co-referent.}

The system uses a logistic regression (MaxEnt) model to make this prediction. As we discussed in class, logistic regression models require a set of \textit{features} to be associated with each training data point. Here, the training data points are \textit{pairs of mentions} that are labeled as either co-referent or not. Given the labeled training data and the set of features for each mention pair, the system will learn a set of \textit{weights} for each feature that optimize its prediction accuracy on the training data.\footnote{Our system also includes a regularization term to avoid overfitting the training data.}

We have already defined two features that can be useful in determining whether two mentions (e.g., \textit{John Smith} and \textit{Mr John Smith}) are co-referent:

- \textbf{MT\_SIM}: This feature measures the similarity between the mention strings themselves. It does so by computing the Levenshtein (string edit) distance between the two mentions (in this case, 3; see J&M 3.11 if you don’t remember the definition of Levenshtein distance).

- \textbf{DOC\_SIM}: This feature measures the similarity between the documents in which the mentions occur. It does so by computing the cosine similarity between two vectors representing the documents: each vector consists of the words in each document, weighted by \textit{tf-idf} (to be discussed in class on Tue 11 Nov, or see J&M 23.1.2).

Your task will be to change these features or add new features that will help determine whether two mentions are co-referent. It is possible to do so with almost no coding, if you so wish, because we have already implemented a number of similarity measures for you to use, e.g., cosine similarity and Jaccard similarity (defined in your textbook). We also provide you with code that extracts various attributes of each mention, such as the text of the document it occurs in, and other attributes described below. What you will need to do is think about which attributes might be useful and what similarity measure(s) might be sensible to use with these attributes. Our code documentation describes how to use the attributes and similarity measures to compute the features to send to the regression model.

\textbf{Note:} we use the term \textit{feature} to refer to features of mention \textit{pairs} that are used as input to the logistic regression model. We use the term \textit{attribute} to refer to properties of individual mentions that could be used to compute features of mention pairs (e.g., by using the attributes of two mentions to compute a similarity score between them, and using this score as a feature).
3.2 Dataset and preprocessing

We use a portion of the Wikilinks dataset \(^4\) (Singh et al., 2012) to give us a set of documents with labeled examples of co-referent mentions. These labeled examples were created largely automatically by finding web pages that contain links to Wikipedia articles. If two pages link to the same article, the text of the link is taken to be the mention text, and the two mentions are assumed to be co-referent. Some filtering was applied to remove links that are likely to be incorrect, but it is possible that some incorrect mentions or mention pairs remain. Nevertheless, the dataset is good enough quality that it can be used to train a system like ours.

So that you don’t have to spend hours training your system on this very large dataset, we are using only a subset of the data here. We chose a set of ambiguous person names (e.g., John, Emily, Clare) and sampled mentions that contained one of the ambiguous person names then further sampled pairs of mentions to create mention pairs for you to train/test your system on.

The data provided for you has been preprocessed using tools available from the Stanford NLP group.\(^5\) We computed POS tags, named-entity tags,\(^6\) and within-document co-reference chains. Since all three annotations are automatic predictions from the Stanford tools, they are sometimes incorrect (especially the within-document co-reference chains).

Finally, we automatically aligned the mention text from the Wikilinks data to the co-reference chains produced by the Stanford tools. This step was needed because the text in the link mention might not exactly match any of the mentions in the co-reference chain extracted from the document that link points to. For example, the link text radio actor John Smith might point to a page where the co-reference system extracted a chain that includes only John Smith, he, and Smith. Given the set of co-reference chains in the document linked to by the mention text, our alignment attempts to match the mention text to the chain it actually refers to. However, this step again could introduce some errors.

3.3 System evaluation

For each mention pair in the test set, our CDCR classification system predicts the probability of that mention pair being linked (co-referent). To evaluate the system, we can choose some probability threshold \(p\), and say that all pairs whose link probability is greater than \(p\) are predicted to be linked. All other pairs are predicted to be not linked. Based on these predictions, we can compute precision (proportion of the predicted links that are correct) and recall (proportion of the correct links that are predicted).\(^7\)

We include code (described below) that computes the precision and recall of the system for values of \(p\) between 0 and 1, and plots the result as a curve like the one in Figure 1. Low values of \(p\) will cause the system to predict more links, normally leading to higher recall but lower precision; high values of \(p\) will have the opposite effect. The plot does not show \(p\) directly, but plots precision against recall. (This particular plot—created using the MT SIM feature alone—has a rather unusual shape for a precision-recall plot, which is related to the particular feature being treated.) Most other systems, which treat CDCR as a clustering task, use more complex evaluation measures.

\(^4\)http://www.iesl.cs.umass.edu/data/wiki-links
\(^5\)http://nlp.stanford.edu/software/
\(^6\)For each word in the document, its NE tag indicates whether it is part of a named entity and if so, what type (person, organization, etc.). In practice our code only cares whether each word is in a named entity or not.
\(^7\)Most other systems, which treat CDCR as a clustering task, use more complex evaluation measures.
used and the structure of our dataset. If you figure out the reason for this shape, you could include that explanation in your report.)

To compare systems, one can either plot precision-recall curves for each system on the same set of axes, or just choose the value of $p$ for each system that produces the highest $F_1$-measure, and compare those $F_1$-measures. Our code computes the highest $F_1$-measure for the system being evaluated and plots it as shown in Figure 1.

### 3.4 How to get started

Once you have familiarized yourself with this document and read through the code documentation (which describes the attributes and similarly measures available to you), here are some things you could try initially:

- **You might want to start by considering the two features that we already implemented.** What are these features intended to do? Do you think they are the best way to achieve that, or are there other features (using different attributes or different similarity measures) that might work better? Look at the plots of the two features individually, and what happens when you use both.

- **You might instead consider other kinds of features that capture different aspects of similarity.** For example, look at the attribute `win_NE_words`, which contains the set of named entity words that occur within a 55 word window of the mention. How could you sensibly
compute a similarity feature based on this attribute, and would you expect adding this feature improve the system’s performance? Why or why not?

You may also have some other ideas for useful features from discussion in lecture or in the textbook, but we would like you to do at least a little bit of additional outside reading as a way to get other ideas and to justify your experiments. Here are a few papers that could be useful for suggesting features to try, or justifying your choice of features.

You do not necessarily need to read these papers in great detail or understand everything they did in order to get some useful ideas from them. Focus on the features they use and the kinds of experiments they run as a way to get ideas about the kinds of features and comparisons you could try, and how to present such experiments.

Full references (including links for download) are in the References section.

• Cohen et al. (2003): This paper describes and compares a large number of string and vector similarity measures, several of which are available to you in the code we provide. Note, however, that their definition for soft-tf-idf is incorrect; the correct definition can be found in Moreau et al. (2008).

• Gooi and Allan (2004): The main thing to look at in this paper is figure 6, showing how different window sizes effect CDCR performance.

• Chen and Martin (2007): This paper describes a number of potentially interesting features for CDCR. Unlike our regression system here which learns weights for each feature, they combine the features using a very simple averaging. (As in the Cohen et al. 2003 paper, the definition for soft-tf-idf is incorrect so refer to Moreau et al. (2008) for that.)

• Lee et al. (2012): This paper describes a more complex system that handles co-reference involving not only entities but also events. Some of their features are for the joint entity and event co-reference task, but they also introduce some features that may be useful for you in doing entity co-reference by itself.

4 Running the code and adding features

We have written some separate documentation giving details of what our code does and how to use it; you will need to read through that documentation to understand how to add features to the system and run your experiments.

5 Writing your report

5.1 What to include in your report

You should turn in a report of no more than six pages (see below) that covers the following four aspects of your work:

• Background research: what did you read or do to think up new features?
• Description of features: what features did you implement and why did you expect they would help?

• Experimental results: comparisons between the results using different feature sets.

• Discussion and conclusion: Did your features work as you expected? Did anything unexpected happen? What insights (if any) can we gain from your experiments and what are the main conclusions you can draw? Are there any obvious loose ends that you might want to tie up if you had more time?

Your report need not be divided into exactly these sections, but it should be clearly organized so that each of the four aspects can be easily seen.

5.2 Length of report and what not to include

Your report is limited to six pages (not including references), assuming a font size and page style similar to what we use here (11 point font). It should even be possible to produce a high-quality report in fewer than 6 pages; please do not get bogged down in needless description or a drawn out introduction. Also please do not provide detailed summaries of your background reading, a few sentences about what ideas you drew from the paper and how they relate to your similarity features should be enough to convince us that you have read the paper and understood it.

This assignment is extremely open-ended and we are intentionally limiting the length of your report to make sure that you do not go overboard in exploring every possible variation of every possible feature. A good report can focus just on one class of features, for example: ‘How should named entity words be taken into account?’ or ‘Does modelling the distance from the mention help when weighting terms?’ or anything you think you can justify and correctly investigate. A well-justified negative result (e.g.: ‘I thought these features might help because [...] these features didn’t help, I think this is the reason why’) will still get the marks.

5.3 Marking scheme

No marks will be given for system performance, in order to encourage you to explore any feature sets you think will be interesting. A 5% performance improvement with a novel feature set is probably more interesting than a 10% improvement with existing features.

You can do well on this assignment by:

• Including appropriate justification for your choice of features, whether it is from your reading or your own intuition.

• Appropriately citing ideas taken from background reading or other sources.

• Clearly describing the experiments you did and the results you got (including appropriately labeled figures and/or tables).

• Discussing the pros and cons of your approach, what might improve your results (and why), and/or other insights from your results.

• Doing all of the above in a clearly organized and clearly written report.
A good justification/exploration/discussion of one of the two ideas we suggested above would be sufficient for a decent passing mark; to get a high mark you should consider additional features.
We reserve the right to deduct marks for over-long reports.

6 Submitting your assignment

Submit your assignment using the following command on a DICE machine:
submit anlp 3 <file.pdf>
where you replace <file.pdf> with the name of the file containing your report. You do not need to submit your code.

Notes:
• Your report is limited to 6 pages, as noted above.
• Your report must be either a .pdf document or .txt (plain text) file, please convert other files types (eg Word documents) into .pdf format before submitting.
• You can submit more than once. Identically named files will overwrite earlier submitted versions. All submissions are timestamped, so do not submit anything after the deadline—it will be considered late, and receive a mark of 0.
• Make sure you save your report before submitting.

References


