DCIT: Discourse Connectives in Twitter

C. Clayton Violand charles.violand@uni-potsdam.de
Jessica Grasso jgrasso@uni-potsdam.de

21 August 2015

Abstract

Do we need an abstract for something so short? I say no.
Discourse connectives are bla bla. DCIT is a tool written in Python that analyzes the usage of discourse connectives amongst German Twitter data. Bla bla

1 Background and Related Work

Discourse markers or discourse connectives are words that show the presence of a discourse relation in text. These words form a closed class, must meet certain fixed criteria, and share some characteristics of content words. These relations are vital to the coherence of a text, and can be used to analyze content and structure [4].

Something about Twitter here?

Part-of-speech tagging for Twitter data faces many challenges due to the non-standard language common on that platform, including acronyms, missing or shortened words, emoticons, and URLs, in addition to items such as proper names. [5]. We cannot comment on which, if any, techniques were used to overcome these challenges, as the output files were provided to us, but this is a limitation to keep in mind.

This project directly uses DiMLex, a lexicon of German discourse markers developed by the Discourse Lab at the University of Potsdam [4]. From this lexicon we extracted our list of German discourse connectives and additional information about each.

We also used results found in the bachelor thesis of Angela Schneider, titled *Disambiguierung von Diskurskonnektoren im Deutschen* and also completed at the University of Potsdam [1], and a paper published by Schneider and Manfred Stede containing a summery of the same work [2]. These results of this work were directly used as a first step to disambiguate the potentially ambiguous discourse connectives in this project.

2 System Description

DCIT is a tool written in Python that analyzes the usage of discourse connectives in German Twitter data. Given a list of German discourse markers and one or more files containing German-language tweets, the tool counts possible discourse connectives, performs some initial disambiguation on the ambiguous connectives, and re-counts, outputting annotated versions of the tweets.

2.1 Input

- `dimlex.xml` - a lexicon of German discourse markers [4]. At the time of submission the current version was that of 25 July 2015.
- One or more `.xml` files containing tweet threads. This format was provided to us and was not altered for this project. Although not perfect, this data was already cleaned and assumed to contain only German-language tweets from a one-month time period.
- For each file above, a corresponding `.txt` file containing the part-of-speech tagged text of each tweet and the unique ID number of each tweet for identification purposes. For this project, these files were provided to us by Wladimir Sidorenko [6], and are tagged using the STTS tag set [3].

2.2 Output

- for each tweet thread file, a modified version of that file with the following tags added:
  - `has_dc` - with value `True` or `False` corresponding to whether the tweet contains at least one (potential) discourse connective
  - `num_dcs` - the number of (potential) discourse connectives contained in that tweet, and,
  - `dc_location` - for each (potential) discourse connective present, the string index at which it begins in the text.

3 Details

First, the DiMLex lexicon is read in and stored as an object (`get_dcons.py`). Each connective has several important features used in this project:

- continuity - whether the discourse connective has one part (*unterdessen*, *und zwar*) or multiple parts (*um ... willen*, *umso mehr ... als*)
- type - for each part, whether that part is single (*um*, *seitdem*) or phrasal (*umso mehr*, *und zwar*)
- ambiguity - true for the entries that have multiple possible interpretations, either as a discourse connective or another function

The lexicon also contains several additional features that we did not utilize.

Then, the tweets are read in (`get_tweets.py`). Since even one day of tweets is far to much to fit in memory (at least on the machines at our disposal), each tweet is sent through the following pipeline individually, and at no point should more than one file (corresponding to one day) be open simultaneously. Each tweet is represented as an object, which at this point contains only features extracted from the file. These include:

- `id` - unique tweet ID number
- `_original` - original, unmodified tweet text
- `words` - tokenized tweet text
- `raw` - tweet text made lowercase and later further modified as needed

The next step is to determine and save more details about each tweet (`get_matches.py`). This includes searching for all possible discourse connectives, and counting and storing some basic information about how many of each has been found. This information, since it is collected over a large number of tweets, is stored in another object (`get_info.py`) which can be

referenced and output at a later time. In addition to these counts, a list of tuples containing all found (potential) discourse connectives, their ambiguity status, and their location in the string is populated in each tweet object for later use.

Once this basic information has been gathered, the task is to disambiguate the ambiguous discourse connectives, that is, to determine which are truly discourse connectives and which are likely to be performing other functions in the text (`disambiguate.py`). For this task we used Schneider's results, resulting in three categories. Category 0 includes the words from Table 1 of Schneider & Stede (2012) [2], and entirely eliminates words that occur only very infrequently as discourse connectives from the list of possible connectives. Category 1 includes the words from Table 2.2 of Schneider [1] for which the ambiguous status can be resolved using the part of speech of the connective in question. Category 2 includes the words from Table 3.1 of Schneider [1] for which further context with part-of-speech tagging can be used.

After this step, the basic counts from above are repeated (`post_disambiguation_stats.py`), using a much simplified but otherwise similar method as before. Compared with the initial stats, there should now be both fewer potential discourse connectives (as some will have been removed during disambiguation) as well as fewer ambiguous matches.

Finally, these results are written to file (`write_results.py`). Currently, the initial files are edited, in that the tags described above are added. These resulting files can then be used in further analysis.

4 Results

These are preliminary results for a subset of the data, namely the 1st of April.

Pre-disambiguation

Post-disambiguation

5 Known Problems and Incomplete Work

- **Optimization** - the tool is quite slow, and could likely be improved through optimization or potentially the incorporation of existing tool or technologies.
- **Testing** - due to time and memory constraints, we were unable to perform very thorough testing. Given the variety of connectives and diversity of Twitter data, this is a vital step.
- **Additional disambiguation** - Schneider's work includes results for only a small fraction of the discourse markers included in DiMLex. While this includes many of the common discourse connectives, other ambiguous connectives remain. Additional research or other methods could be used to improve disambiguation.
- **Conversations** - the project currently focuses on single tweets, not pairs or conversations among Twitter users. Since tweets are very short texts, and many discourse connectives allow arguments over larger spans, this should be further expanded.

6 Conclusion and Future Work

In addition to the improvements listed above, future work could include...

bla bla something about the stats?

Acknowledgement

Thanks to Paul Ebermann for some suggestions on optimization and general programming advice.

References

[1] Angela Schneider. *Disambiguierung von Diskurskonnektoren im Deutschen* (Unpublished bachelor's thesis). Universität Potsdam.

[2] Angela Schneider & Manfred Stede. (2012). *Ambiguity in German connectives: A corpus study.* In: *Proceedings of KONVENS 2012 (Main track: poster presentations)*, 254-258.

[3] Anne Schiller, Simone Teufel, Christine Stöckelt & Christine Thielen. (1999). *Guidelines für das Tagung deutscher Textcorpora mit STTS (Kleines und großes Tagset).*

[4] Manfred Stede. (2002). *DiMLex: A lexical approach to discourse markers.* In: A. Lenci, V. Di Tomaso (eds.): *Exploring the Lexicon - Theory and Computation.* Alessandria (Italy): Edizioni dell'Orso, 2002.

[5] Uladzimir Sidarenka, Tatjana Scheffler, & Manfred Stede. (2013). *Rule based normalization of German Twitter messages.* In: *Proceedings of the International Conference of the German Society for Computational Linguistics and Language Technology.*

[6] Wladimir Sidorenko. (20 August 2015). University of Potsdam staff home page. `<http://www.ling.uni-potsdam.de/staff/sidorenko>`