1 Introduction

This project deals with using information extraction techniques to capture compatibility relations between software components that exists in the form of unstructured text on online forums.

2 Dataset

My goal was to focus on Q&A sites such as Stack Overflow. Stack Overflow is part of a wider network of Q&A sites known as StackExchange (https://stackexchange.com). While Stack Overflow is a generic developer community, there are sites that focus on specific topics such as Ubuntu, Networking and Security. Among these ServerFault (https://serverfault.com) is a Q&A site for system and network administrators. This site was selected to be the dataset since the posts on here usually deal with a wide variety of software components to be used in different scenarios. Therefore, it is a potential source of information about inter-component relationships. StackExchange offers an online data explorer tool (https://data.stackexchange.com/) where SQL queries can be run in the browser and data saved in the form of a CSV file. However, through this mode, the number of records that can be retrieved is restricted to 50000. The complete data set archives are made available at (https://archive.org/details/stackexchange). The current available set contains all data till September 2018 and is available in the form of XML files. To enable easier data analysis, the XML files were imported into a SQL Server database using the StackOverflow data dump importer tool (https://github.com/BrentOzarULTD/soddi/).

3 Approach

The overall approach is to adopt a supervised learning method. Therefore, to build the training data set we need to identify entities and tag the appropriate relations between them. On further examination of the posts, it is observed that latent incompatibilities are not only manifested between software components (e.g. libraries) but also between a component and a particular feature. E.g. consider the following sentence: The underlying cause for this is that vboxsf does not support filesystem links (neither hard nor symbolic). Here the incompatibility is that the virtual box shared folder filesystem (vboxsf) does not support filesystem links. Here “filesystem links” can be viewed as components if the words are seen separately, but taken together they represent a feature that is not supported by vboxsf. Therefore it is better to capture both component-component and component-feature type of relations. An approach to handle this was to identify nouns or noun-phrases to be the named entities. Relations could be then be extracted between pairs of such entities using techniques such as DNN’s. The task therefore is similar to the SemEval-2010 relation extraction task.

4 Preprocessing and Bootstrapping

Only posts that are answers are decided to be included in the training set. While questions can contain useful information too, answers are more likely to express extractable relations. The dataset consists of a total of 436751 answers. To bootstrap the data and to form a set of reasonable size for manual tagging, some filtering tactics were used. Answers can be narrowed down by the tags associated with their parent question post. Further, the answer body can be filtered on specific keywords such as “supports” or “requires” that can indicate the presence of a relation. Further, the
number of votes of an answer can also be used as a further quality filter. Using the PostgreSQL question tag, the `supports` keyword in the body and a vote of at least one, the number of retrieved answers is 103. To form a training data set similar to the SemEval-2010 Task 8 corpus, the aim is to split each answer body into individual sentences and tag the nominals within the sentence. The body of the posts contains text in the form of rendered HTML as it appears on the site. Therefore, the HTML tags and any code snippets were cleaned from the body. The post was then split into sentences and nominals were tagged.

5 Named Entity Tagging

The Spacy (https://spacy.io) library was used for sentence disambiguation and entity tagging. Entities were tagged using the noun chunks that were identified. The following issues were observed:

- Many pronouns such as I, them, they were identified as entities. This was fixed by excluding the pronouns as identified by their Part of Speech Tag.

- Coreference resolution: Related to the above point, using coreference resolution can address the case where if the corresponding reference for the pronoun is identified, it need not be discarded.

6 Relation Extraction Annotation

The aim is to manually label sentences with a class tag for each relation type to extract. The following relation classes are identified:

- Requires: The requires relation indicates that an entity is strongly dependent on the second entity.

- Incompatible: The incompatible relation indicates that the two tagged entities are incompatible with each other.

- Other: This label is for the entities in the sentences that exhibit other relations or are not related.

The following issues were observed in the manual labeling process:

- More than two entities: The noun chunking process produces more than two entities in a sentence. However, for the purposes of relation labeling, we are interested in sentences with exactly two entities. One strategy is to retain only those sentences with exactly two entities. However, in the above filtered example for the `postgres` tag, the total number of sentences was 910. Excluding sentences with less than two entities, the total number of sentences with exactly two entities is about 36%. This means that a majority of the sentences have more than 3 entities and will be discarded. Therefore, this leads to a situation where a ranking of the pairs of entities in a sentence can be performed to retain the top two entities. Zhu et al. (2015) use several features to characterize similarities between Stack Overflow tags. A similar approach could be used to rank candidate entities based on the information in the tags attached to the question.

- Quantity of training labels: Retaining sentences with only two nominals, gives a sentence set of 185. Manual examination showed about 10 sentences that had entities with relations that could be labeled (excluding ‘Other’). Therefore, it could lead to an issue with the training data having disproportionately large number of relations classified as ‘Other’.

7 Summary

To summarize, entity tagging accuracy can be improved over the noun chunking method. This could be addressed with domain specific training. Additionally, coreference resolution and ranking of entity pairs could be used to improve performance. This can lead to increasing the quality of the labeled training set.

References