Wikification of news

By

Mart Busger Op Vollenbroek S2174634
Olivier Louwaars S2814714

Project Text Analysis
ms. M. Nissim
mr. L. Kloppenburg

MART

The project started for us when we had to make project groups. Olivier and I both started this year as Premaster students and we got to know each other (and Chris & Leonardo) better than we got to know the first year students. That is why it wasn't such a hard task to form groups, as four Premaster students we splitted up in groups of two, thinking we could handle a first year project with just the two of us. Seeing as we were at about the same level of programming, we understood each other and came up with almost similar approaches to the problems.

Though the exercises prepared us in a way, but they were not enough to see the final project as an assembly of the weekly exercises. We touched a lot of subjects during the course and many were used during the final project, but we still had to come up with some creative solutions to solve the final project.

During the project I mainly focused on getting the data in the right format and using WordNet. We thought that the way the data was structured (using the columns), made it a bit difficult to work with. It would have been way easier to just have supplied us with texts. Eventually we got the hang of it and that made the whole project a lot easier. Even though it might not be the most efficient option, it worked for us so we were satisfied. The second part to which I contributed a lot was the use of WordNet, especially the use of synsets. We thought those were very confusing at first but again, when we got the hang of it wasn't that hard. Finally I contributed to the final part of the wikification of the tagged words. Some words weren't given a Wikipedia page, so we had to come up with something to make sure that every tagged word had a Wikipedia page. The result of that you can see in annotate.py in the wikiexpander() function and reverseTagset() function.

For further information about our project we would like refer to our Github repo:

https://github.com/Martbov/pta-group1

OLIVIER

The moment Mart and I started this project, we approached it as a team. Being in one class for ¾ year now, we both know what we can do ourselves and what the other is capable of too. Being on similar levels of programming, the dividing off the tasks was not really necessary. Each one of us would take a turn to write some code, while the other was watching and thinking about the problems ahead. Taking turns resulted in a program we both support and understand, and we would have (eventually) reached similar programs if we would have to do it on our own. Although the exercises for PTA prepared us a bit for the final assignment, it still was hard to combine all the small programs. Mostly due to the strict demands in formatting and order, using unordered data containers as sets and dictionaries was not ideal. From time to time we felt like we missed the knowledge, which probably has cost us a lot of time and workarounds.

During the final project, I focused mainly at the machine learning part. During the course information retrieval, we already learned how classifiers work which helped me here. As the classes are not commonly used in other classifiers, we had to start from scratch and build our own. Knowing the Stanford NER tagger is one of the best available, training it on our data had the best perspective. Using a online tutorial, it was fairly easy to do so, and it seems to classify with great performance. Of course this is only on the given train data, so it still has to proof itself on the real data. The hardest part was the wikifying itself, after the important entities were recognized. The method we eventually used was to look up the entity in WordNet for a definition, and then scanning the Wikipedia API suggestions for the same words. Although this seems a logical approach, the machine thinks different and comes up with very strange URL's sometimes (still always related to the original entity). It will always be hard for a machine to achieve the same results as humans because of the lack of context that makes human decide to link to a certain page.