**Sprint Evaluation**

In this sprint, our team was planning to complete sentence representation compressing and feature evaluation part. As we wrote in the start-up document, we planned to complete summarization part of whole project in this term. At the end of this sprint, we are really close to complete what we planned. First, we started to finish our backlog task from previous sprint, which is sentence representation and feature evaluation. We almost completed these import tasks. Beside that, we started to use our Deep Learning elements which are auto-encoders and classifiers (Neural Networks). Our research session took to much time because design is very important at this step. We searched for different implementations and decided which programming languages is more useful for this task. We found Python and Matlab implementations, Matlab is more useful for auto-encoders. Also, there are two types of auto-encoders we can used, which are sparse and dense. We decided to use  sparse auto-encoders. Simultaneously, we completed our sentence representation and feature evaluation. This step is took 2 sprint, because there are so many works to do as we are not mentioned in start-up document. We used Tensorflow (Python), Zemberek (Java) and wrote our own scripts for specific features in Python.

We also noticed that our data texts are not suitable for summarization. We searched for new text which are suitable for summarization and our project.

**Team evaluation**

We had weekly meetings with our assistant and we also met each weekend working on our project, brainstorming how can we improve it. In this sprint, we started to work as a team more, since the individual work is finished. We have started to compound our work.

| Task | Assigned Member | 1st week | 2nd week | 3rd week |
|---|---|---|---|---|
| Using Zemberek tools to get sentence features | Abdullah Göktuğ MERT | √ | √ | |
| Research for Auto-Encoder Implementation | Baran Barış KIVILCIM, Enes Uğur ŞEKERCİ | √ | √ | |
| Implementing Auto-Encoders in Python | Enes Uğur ŞEKERCİ | | √ | |
| Implementing Auto-Encoders in Matlab | Baran Barış KIVILCIM | √ | √ | √ |
| TRNLTK tokenizer and improve sentence parsing | Abdullah Göktuğ MERT | | √ | |
| Getting all features together | Yağız ARKAYIN | | √ | |
| Review features and bugfix | Yağız ARKAYIN | | √ | |
| Training Word2Vec Data Corpus | Enes Uğur ŞEKERCİ | | √ | |
| Centrality Feature Implementation | Baran Barış KIVILCIM | | √ | |
| Training Auto-Encoders | Enes Uğur ŞEKERCİ, Baran Barış KIVILCIM | | √ | √ |
| Finding new texts | Yağız ARKAYIN | √ | | |
| Obtaining data from real users | All team | √ | √ | √ |
| Class Representation | Yağız ARKAYIN, Baran Barış KIVILCIM, Abdullah Göktuğ MERT | | | √ |
| Research for classifiers | Baran Barış KIVILCIM | | | √ |
| Implementing classifiers | Baran Barış KIVILCIM | | | √ |

|  |
| --- |
|  |

**Backlog Updates**

In the second sprint, we couldn't finish sentence representation, but in this sprint we have finished it. As our project requires as more data as possible, we can't say we have finished gathering data from people. As we stated in the start-up document we will continue to do so in the next sprints.

The algorithm we planned to implement is finished, but our supervisor mentioned another algorithm for the classifier phase. So, we haven't completed this new algorithm. In the next two weeks we plan to finish implementing the algorithm and combine these two algorithms to have better results.

Apart from these two issues, we have completed what we have planned in the start-up document. However, most of these parts are still independent and based on different programming languages. In the next sprint, we plan to combine all these parts under the same programming language or a script.