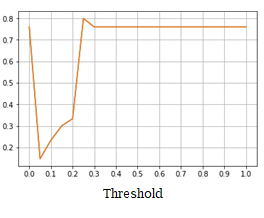
**Group 12**

Gonçalo Lopes 76213, Adriana Pereira 75384, Jessica Ramgi 75619

1. **Simple graph-ranking approach**

For the ranking procedure we used a maximum number of 50 iterations and a damping parameter of d = 0.15. The minimum threshold of cosine similarity between sentences was varied, using values between 0 and 1 with an interval of 0.05 between values.

For the text we chose a review of a TV show, and produced an “optimal” summary using an online summarization tool.

The obtained results are shown on Figure 1.

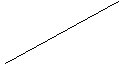
The highest Average Precision was of 0.8, for a value for the threshold of 0.25.



1. **Improved graph-ranking**

|  |  |  |
| --- | --- | --- |
| Weight  Prior | Uniform | Cosine Similarity |
| Uniform | 0.1598 | 0.16733 |
| Degree Centrality | 0.163 | 0.1695 |
| Sentence Position | 0.1694 | **0.1722** |

*Figure 2: MAP for the different combinations of prior probabilities and weights.*

The obtained results are shown on Figure 2.

We implemented the following prior probabilities:

* Degree Centrality – sentences with more nodes in a graph are more important.
* Sentence Positions – sentences that appear first in a document are more relevant.

For the weights we used the cosine similarity between sentences.

We also considered removing stopwords, but obtained worse results, with a maximum MAP of 0.1695.

Sentence position and Cosine Similarity yielded the best results, improving the simple approach (uniform prior probabilities and weights) by 0.124.

1. **Learning-to-Rank**

For this approach we used the Perceptron algorithm, considering as features the position of a sentence in the document, the cosine similarity between each sentence and the document, degree centrality as implemented for the second approach and a naive-Bayes classifier. For this last feature we tested classifiers based on Gaussian, Bernoulli and Multinomial distributions, but found that, all else being equal, the MAP scores were the same for models obtained with the Bernoulli and Multinomial classifiers, and these scores were better that the score gotten assuming Gaussian distribution. Given that the features we are using are not binary, we chose to use the naive-Bayes classifier that assumes multinomial distribution. We also found that using logarithmic probabilities seems to be better. In order to obtain better results, we are removing stopwords on this approach when calculating the cosine similarity values.