Content Based Similarity Matching of BPMN models

09.12.2020

Patka Zsolt-András

1 Introduction

The main idea of my approach was to convert the BPMN processes to texts, because a lot of approaches already exist for comparing two texts semantically [1]. I also wanted to keep things simple and offload the text similarity logic to an already existing and well functioning API. For this I used the Twinwords API.

2 Difficulties

Unfortunately Twinwords does not provide much description about how their similarity calculations are performed, what algorithms are used, and what exactly the return value means. The most that they write about their algorithms is the following: "If you simply want to know the relationship score between two documents, just make an API call and you got it." To get a deeper understanding of the return values, I had to manually test the API with multiple different text pairs and observe the return values.

3 Exploration of the text similarity algorithm

I conducted some tests in order to explore the algorithm behind the Twinwords Text Similarity Matching API and to get a deeper understanding of the return value.

Test case: Same texts
Text 1: Once upon a time in Holywood
Text 2: Once upon a time in Holywood
Result: 1
Observation: Identical texts have a relationship score of 1.

Test case: Synonyms
Text 1: angry
Text 2: enraged
Result: 0.77
Observation: Synonyms can be recognized.

Test case: Antonyms
Text 1: good
Text 2: bad
Result: 0.22
Observation: Antonyms have lower score.

Test case: Same topic, different categories
Text 1: PC
Text 2: Mac
Result: 0
Observation: Categories are not always picked up correctly.

Test case: Same words, different language
Text 1: almás tészta
Text 2: apple cobbler

Result: 0

Observation: It is not able to translate words. It also does not give a negative score in case the words are in a different language.

Test case: Two words in Hungarian

Text 1: tészta

Text 2: pite

Result: 0

Observation: It can not compare words in Hungarian.

Test case: Two similar texts in hungarian

Text 1: almás pite

Text 2: almás tészta

Result: 0.5

Observation: In this case it compares the two terms in a lexical manner. The first two words are identical, half of the text matches, that is why the score is 0.5

Test case: Two BPMN Processes.

Text 1: (Application processing) File the correspondence. Application received. Post the policy documents. Non-exclusive Gateway. Non-exclusive Gateway. End Event. Issue the policy. Print the address.

Text 2: (Loan increase application) End Event. process application. Check customer file. Contact the customer. Is the application complete?. Is the signature valid?. Examine the application. Check the signature. receive application. yes. no. no. yes.

Result: 0.62

Observation: These two processes are somewhat similar, as both are about application processing. They have identical words (file, process, application, receive) and these were identified by the algorithm.

Test case: Two BPMN Processes.

Text 1: (Application processing) File the correspondence. Application received. Post the policy documents. Non-exclusive Gateway. Non-exclusive Gateway. End Event. Issue the policy. Print the address.

Text 2: (Car re-allocation) 'Reason?. Is a new car needed?. Examine the reason for reallocation. End Event. Cancel because of company's decision. Cancel due to the contract being up. Cancel due to accident. Allocate a new car. Receive car reallocation claim. contract is up. no. decision by the company. yes. car was in an accident.

Result: 0.38

Observation: These two processes are not that similar, they do not contain that many identical words. And there are few words which belong to the same category.

3.1 Conclusion

The algorithm returns values between 0 and 1 where 1 means that the texts are identical and where 0 means that the texts are completely different. A score of 0 is produced when the input texts are in different languages or both of them are not in English. The algorithm can recognize synonyms and words that belong to the same topic.

The algorithm is invariant to the words' order.

4 Possibilities

The provided services could be used for sorting BPMN processes based on their content.

References

[1] W. H. Gomaa, A. A. Fahmy, *et al.*, "A survey of text similarity approaches," *International Journal of Computer Applications*, vol. 68, no. 13, pp. 13–18, 2013.