Data Preprocessing and Text Mining

2023/2024

Julia Cwynar

68846
Universidade Nova de Lisboa
Faculty of Science and Technology
j.cwynar@campus.fct.unl.pt

Andrea Cantelli

69609
Universidade Nova de Lisboa
Faculty of Science and Technology
a.cantelli@campus.fct.unl.pt

Micol Curci

69675
Universidade Nova de Lisboa
Faculty of Science and Technology
m.curci@campus.fct.unl.pt

1. INTRODUCTION

1.1 Project Objective

The primary goal of this text mining project is to analyze a collection of text documents to identify the most significant relative expressions using the LocalMaxs extractor. This algorithm employs three different cohesion metrics - Symmetric Conditional Probability (SCP), Dice Coefficient, and φ^2 (phi-squared) - to extract word combinations that occur together frequently and with a higher level of significance in the context. The focus is on n-grams with a maximum length of 7 words, and only those that appear at least twice without special characters and stop words at the beginning or at the end of the expression can be considered as relative expressions.

1.2 General Project Execution

The project is carried out in several key phases:

1.2.1 Extraction of Relative Expressions:

- LocalMaxs Algorithm: We implemented this algorithm in Python to extract relative expressions using the SCP, Dice, and φ^2 metrics.
- N-grams Generation: We generated n-grams up to a maximum length of 7 words.
- Cohesion Metrics Calculation: These metrics were applied to identify the most significant word combinations, since we know that an expression can be considered relevant only if the cohesion metric is higher than its neighbors n-gram.

1.2.2 Identification of Explicit and Implicit Keywords:

- Explicit Keywords: We extracted the most 10-15 informative relative expressions of each document and considered them as explicit keywords of the document.
- Implicit Keywords: Using advanced NLP techniques such as semantic analysis and word embeddings, we identified keywords based on context and semantic relationships.

1.2.3 Evaluation of the algorithm:

- Precision, Recall, and F-Metric: These metrics were calculated to evaluate the effectiveness of the cohesion metrics used by the LocalMaxs algorithm.

1.3 Theoretical Concepts

1.3.1 Cohesion Metrics:

- Symmetric Conditional Probability (SCP): This metric measures the probability of words appearing together relative to their individual occurrences, highlighting their interdependence.
- Dice Coefficient: A similarity measure comparing the co-occurrence of words to the sum of their individual occurrences, useful for identifying strongly correlated word pairs.
- φ^2 (phi-squared): A statistical measure that evaluates the strength of the association between words using contingency analysis.

1.3.2 Evaluation Metrics:

- Precision: Indicates the proportion between the correct relevant expressions and the total of the relevant expressions found by the algorithm.
- Recall: Measures the algorithm's ability to identify all relevant expressions present in the text.
- F-Metric: The harmonic mean of Precision and Recall, providing an overall assessment of the algorithm's performance.

1.4 Advantages of the Analysis

Conducting this type of text mining analysis offers numerous benefits. One of the most notable advantages is content summarization. By identifying the most significant expressions in a text, we can grasp its core content without the need to read it in its entirety. This is particularly beneficial for quickly gaining insights, as it allows for an efficient overview of the main themes and ideas present in the text.

Another significant benefit is the efficiency in data analysis. This approach substantially reduces the time and resources required to analyze large volumes of text. Instead of manually sifting through extensive documents, the algorithm quickly highlights the most relevant parts, streamlining the process and making it more manageable.

The analysis also provides valuable semantic insights. By examining the relationships between words, we can uncover deeper meanings and contexts that are not immediately obvious. This is useful for contextual analysis and gaining a more profound understanding of the text's nuances and underlying themes.

Moreover, the insights gained from this type of analysis have practical applications in various fields. For instance, in marketing, understanding key expressions can help tailor content and strategies to better resonate with target audiences. In media analysis, it allows for a more precise identification of trends and significant topics. Academic research benefits from the ability to quickly extract essential information from vast amounts of literature, while information management can be optimized by efficiently organizing and categorizing content based on the most relevant expressions.

2. DATASET

The whole dataset that the project is based on, consists of 3170 text files. In total, the number of words combined equals 2209654. Given text files are extracted fragments of longer pieces of text. All brought together, they create a whole corpus.

Greek Christian scribes played a crucial role in the preservation of Aristotle by copying all the extant Greek language manuscripts of the corpus. The first Greek Christians to comment extensively on Aristotle were John Philoponus, Elias, and David in the sixth century, and Stephen of Alexandria in the early seventh century. John Philoponus stands out for having attempted a fundamental critique of Aristotle's views on the eternity of the world, movement, and other elements of Aristotelian thought. After a hiatus of several centuries, formal commentary by Eustratius and Michael of Ephesus reappears in the late eleventh and early twelfth centuries, apparently sponsored by Anna Comnena.

ASCII codes represent text in computers, communications equipment, and other devices that use text. Most modern character-encoding schemes are based on ASCII, though they support many additional characters.

Figure 1. Sample of corpus

3. DATA PREPROCESSING

To enable relevant and efficient research on a given dataset, it had to go under process of tokenization, which means breaking down a text into smaller components.

In our case, we separated each word of each document into separate elements by adding space character before and/or after characters listed below:

```
",;:!<>()[]&?=+\"'./\\"
```

This has given us the result of 2209654 words in the whole corpus, within 3170 documents.

Christian scribes played a crucial role in the preservation of Aristotle by copying all the extant Greek language manuscripts of the corpus . The first Greek Christians to comment extensively on Aristotle were John Philoponus , Elias , and David in the sixth century , and Stephen of Alexandria in the early seventh century . John Philoponus stands out for having attempted a fundamental critique of Aristotle ' s views on the eternity of the world , movement , and other elements of Aristotelian thought . After a hiatus of several centuries , formal commentary by Eustratius and Michael of Ephesus reappears in the late eleventh and early twelfth centuries , apparently sponsored by Anna Comnena .

ASCII codes represent text in computers , communications equipment , and other devices that use text . Most modern character-encoding schemes are based on ASCII , though they support many additional characters .

Figure 3. Sample of preprocessed corpus

4. EXTRACTING RELEVANT INFORMATION

To maintain all relevant information for each of the extracted ngrams, we created a nested dictionary. Each of the elements consisted of the ngram as a key and the values listed below. Values listed below were calculated during the runtime of the program asynchronously.

- Counter - number of occurrences of ngram
- Scp - scp metric of ngram
- On+1_scp - scp metric of ngram after indicated one
- On-1_scp - scp metric of ngram before indicated one
- Dice - dice metric of ngram
- On+1_dice - dice metric of ngram after indicated one
- On-1_dice - dice metric of ngram before indicated one
- Phi2 - phi square metric of ngram after indicated one
- On+1_phi2 - phi square metric of ngram after indicated one
- On-1_phi2 - phi square metric of ngram before indicated one
- Doc_with_ngram - number of documents containing ngram
- Freq_per_document - number of occurrences of ngram with
- Document
- Word_total
- tf-idf

5. EVALUATING NGRAMS

Ngram is a contiguous sequence of n words from a given sample of text.

We calculated all the possible ngrams of length ranging from n=1 to n=7 for each of the documents of the corpus.

Subsequently, we counted the occurrence of each ngram in each document by creating a dictionary with a n-gram as key and counter as value.

For each of the documents we counted the occurrence of a specific ngram among the whole corpus and assigned the value of occurrence to a specific document for each of the ngrams. This

way we could keep track where exactly specific ngram reside beside its primary occurrence.

6. EVALUATING STOP WORDS

Stop words are common words in a language of little value due to its frequent occurrence and no significant semantic meaning.

By applying removal of stop words we could remove additional noise.

We calculated the biGramNeigh values, which are the number of unique neighbors (previous and next words) for each potential stop word, which would be the middle of a trigram. Later it was adjusted by the number of syllables in a specific ngram.

□□□□□□□□□□□*h*(□)
= □□.□□□□□□ □□□□*h*□□□□ □□ □

We approximated the number of syllables in each word by counting the number of vowels.

□□□□□□□□□(□)
= □□.□□□□□□
− □□.□□□□□□□□□□□ □□□□□□

To determine the stop words we calculated neigSyl(w) metric.

$$\square\square\square\square\square\square\square(\square) = \frac{\square\square\square\square\square\square\square\square\square\square h(\square)}{\square\square\square\square\square\square\square\square\square\square(\square)}$$

Afterwards, we sorted the values in decreasing order and extracted the stop words using heuristic below:

$$\square = \square\square\square\{\square \mid \square\square\square\square\square(\square) > \square\}$$

value(r) - neighSyl at rank r

Evaluation of metric b equal to 254, returned 254 stop words, which were 0.01% of the whole corpus. Later were removed from the list of ngrams to increase the performance of our algorithm and return more relevant output.

[',', '.', "", 'and', 'the', 'of', 'in', '(', 'to', 'a', ')', 'was', 'is', 'for', 'The', """, 'by', 'with', 'as', 'on', 'from', 'at', 'that', '=', 's', 'or', ':', 'his', 'an', ';', 'their', 'has', 'had', 'which', '/', '>', 'its', 'are', 'were', 'he', 'who', 'it', 'her', 'first', 'be', 'two', 'but', 'this', 'when', 'In', 'not', 'A', 'de', 'would', 'new', 'through', 'all', 'can', '-', '&', 'also', 'being', 'been', '–', 'then', 'they', 'three', 'will', 'both', 'John', 'may', 'have', 'out', 'him', 'than', 'This', 'into', 'such', 'after', 'most', 'between', 'up', 'no', 'she', '<', 'other', 'He', 'them', 'team', 'during', 'four', 'could', 'more', 'each', 'I', 'one', 'near', 'over', 'against', 'only', 'now', 'about', '2', 'group', '1', 'under', 'so', 'some', 'small', 'played', 'school', 'won', 'took', 'still', 'His', 'film', 'off', 'work', 'own', 'did', 'It', 'until', 'while', 'School', 'French', 'left', 'many', 'high', '3', 'years', 'where', 'if', 'New', 'band', 'World', 'main', 'found', 'back', '4', 'made', 'later', 'due', 'year', 'known', '5', 'last', 'these', 'just', 'set', 'long', 'called', 'show', 'down', 'since', 'around', 'Park', 'High', 'any', '000', 'died', 'via', 'County', 'what', 'Street', 'must', '10', 'They', 'per', 'held', 'people', 'club', 'West', 'said', 'much', 'began', 'II', 'six', 'went', 'Road', 'before', 'like', '!', 'became', 'North', 'old', 'best', 'season', 'Group', 'including', 'May', 'great', 'men', 'form', 'used', 'less', 'should', 'When', 'South', 'part', 'led', '6', 'world', 'system', 'local', 'former', 'play', 'Act', 'third', 'land', 'player', 'song', 'without', 'time', 'On', 'within', '7', 'She', 'British', 'law', 'short', 'teams', 'points', 'lost', 'Paul', 'works', 'do', 'early', 'River', '?', 'War', 'Cup', 'there', 'second', 'book', 'playing', 'good', '8', 'full', 'soon', 'few', '—', 'several', 'built', 'run', '2011', 'King', 'next', '12', '20', 'young', 'March', 'state', 'players', 'An']

Figure 3. Sample of stop words list

7. COHERENCE

Having calculated a list of n-grams of the whole corpus subtracted by stop words, we wanted to measure coherence of each expression. As some of them combine with a specific set of words by forming cohesion groups, we wanted to find the most relevant expressions within the whole corpus.

Relevant expression is a sequence of n-grams within a corpus that holds significance in terms of cohesion and semantic relevance. They tend to be indicative of specific topics within the text.

To evaluate which of the expressions holds the biggest significance, we calculated 3 metrics and for each of n-grams we updated the dictionary with its values.

7.1 Symmetric Conditional Probability coefficient

$$\square\square\square((\square,\square)) = \square(\square|\square).\square(\square|\square) = \frac{\square(\square,\square)^2}{\square(\square).\square(\square)}$$

$$\square\square\square_{\square}(\square_1,..,\square_{\square}) = \frac{\square((\square_1,\ldots,\square_{\square}))^2}{\square}$$

$$\square = \frac{1}{\square - 1} \sum_{\square=1}^{\square=\square-1} \square(\square_1, \ldots, \square_\square, \ldots, \square_\square)$$

7.2 Phi-square coefficient

$$\phi^2((\square, \square)) = \frac{(\square \times \square(\square, \square) - \square(\square) \times \square(\square))^2}{\square(\square) \times \square(\square) \times \square(\neg\square) \times \square(\neg\square)}$$

$$\square(\neg\square) = \square - \square(\square)$$

$$\phi^2{}_\square((\square_1, \ldots, \square_\square)) = \frac{\square \times \square((\square_1, \ldots, \square_\square)) - \square\square\square)^2}{\square\square\square}$$

$$\square\square\square = \frac{1}{\square-1} \sum_{\square=1}^{\square=\square-1} \square(\square_1, \ldots, \square_\square) \times \square(\square_{\square+1}, \ldots, \square_\square) \qquad \square\square\square = \frac{1}{\square-1} \sum_{\square=1}^{\square=\square-1} \square(\square_1, \ldots, \square_\square) \times \square(\square_{\square+1}, \ldots, \square_\square).$$

$$(\square - \square(\square_1, \ldots, \square_\square)) \times (\square - \square(\square_{\square+1}, \ldots, \square_\square))$$

7.3 Dice coefficient

$$\square\square\square\square((\square, \square)) = \frac{2 \times \square(\square, \square)}{\square(\square) + \square(\square)}$$

$$\square\square\square\square_\square((\square_1, \ldots, \square_\square)) = \frac{2 \times \square((\square_1, \ldots, \square_\square))}{\square}$$

$$\square = \frac{1}{\square - 1} \sum_{\square=1}^{\square=\square-1} \square(\square_1, \ldots, \square_\square) + \square(\square_{\square+1}, \ldots, \square_\square)$$

Those are the results we obtained using these metrics:

N° of RE found with SCP metric: 16020

N° of RE found with Dice metric: 26384

N° of RE found with phi2 metric: 16017

As expected the number of RE found with the Dice metric is way higher than with the others.

8. EVALUATING METRIC

8.1 Precision

The first function we define to assess the accuracy of our algorithm is precision. This involves randomly selecting 200 relevant expressions from those identified by the algorithm and evaluating, based on personal judgment, which of these can be considered truly significant. The result of the function is determined by dividing the number of personally chosen relative expressions by the 200 relative expressions selected randomly.

This Precision function is applied to each of the three metrics used in our project, resulting in three different precision values.

Precision with SCP metric: 175/200=0.875

Precision with Dice metric: 165/200=0.825

Precision with phi2 metric: 150/200=0.75

8.2 Recall

The second function we use to evaluate the algorithm is called recall. As a first step, we read and analyzed 10 paragraphs from the text file and compiled a vector of 200 expressions that we deemed significant. Subsequently, we compared the expressions found by us with those identified by the algorithm, and the number of common expressions was divided by 200 to obtain the recall value. This function was also applied to all three metrics, resulting in three different recall values.

Recall with SCP metric: 7/200=0.035

Recall with Dice metric: 8/200=0.04

Recall with phi2 metric: 7/200=0.035

8.3 F-metric

$$\square_{\square\square\square\square\square\square} = \frac{2 \square \square\square\square\square\square\square\square\square \square \square\square\square\square\square\square}{\square\square\square\square\square\square\square\square\square\square + \square\square\square\square\square\square}$$

F_metric with SCP metric: 0.0673

F_metric with Dice metric: 0.0763

F_metric with phi2 metric: 0.0669

8.4 Conclusions

The SCP metric has the highest precision (0.875), indicating that the relevant expressions are more likely to be significant.

The Dice metric has sligthly the highest recall metric (0.04) compared to two others (0.035). This low value indicates that many significant expressions may be missed.

F-metric for Dice metric has the highest value (0.0763) indicating a better balance between precision and recall.

The Dice metric identified significantly more relevant expressions (26,384).

Looking at these values we can say that the relative expressions found with the Dice metric can be considered as more valuable since the algorithm performs better using it as a metric to obtain meaningful relevant expressions.

9. EXPLICIT KEYWORDS

Explicit keywords are terms or phrases that are directly mentioned in the text. They are straightforward and do not require any interpretation to be identified.

To evaluate them we applied tf-idf measure for the ngrams chosen before

Tf-idf is a statistical metric consisting of:

- Term Frequency

$$\square\square = \frac{\square\square\square\square(\square\square, \square\square)}{\square\square\square\square(\square\square)}$$

size(dj)) - number of words of di
freq(RE,dj) - frequency of RE occurrence in dj

- Inverse Document Frequency

$$\square\square\square = \square\square\square\left(\frac{||\square||}{||\{\square \in \square \wedge \square\square\square\square(\square\square, \square) > 0\}||}\right)$$

||D|| - number of documents of corpus D
{□ ∈ □} - number of documents containing term

After calculating this tf-idf, we chose 15 of the most explicit keywords within each document, it is, 15 n-grams per each document with the highest tf-idf metric.

10. IMPLICIT KEYWORDS

Implicit keywords are terms or concepts that are not directly mentioned in the text but are inferred from the context, associations, or underlying themes.

Having the list of the relevant expressions, we created a mutual set consisting of the relevant expressions chosen before and additionally single words, approximating them by taking the first and the last word of each relevant expression.

The output set after excluding all repetitive elements consisted of 27869 elements.

With this set, we calculated the Inter Document Proximity, also known as correlation, for each of the indicated elements.

This metric refers to the closeness or similarity between terms A and B across multiple documents. It measures how often and how closely terms appear together in different documents.

We constructed a dictionary with as key a pair of words from the set and explicit keywords from documents, and with the correlation metric as the value. This has helped us also to not calculate

multiple times without reason the same correlation metric, saving some execution time.

$$\square\square\square\square(\square,\square) = \frac{\square\square\square(\square,\square)}{\sqrt{\square\square\square(\square,\square)} \times \sqrt{\square\square\square(\square,\square)}}$$

$$\square\square\square(\square,\square) = \frac{1}{||\square|| - 1} \times \square(\square(\square,\square\square) - \square(\square,.)) \times (\square(\square,\square\square) - \square(\square,.))$$

$$\square(\square,.) = \frac{1}{||\square||}\square\square(\square,\square\square)$$

$$\square(\square,\square\square) = \frac{\square\square\square\square(\square,\square\square)}{\#\square\square\square\square\square(\square\square)}$$

High correlation indicates that terms A and B frequently appear together across different documents, suggesting a strong relationship or common theme involving both terms.

Having calculated the correlation between each of the most explicit keywords evaluated before and all the terms of the set, we wanted to calculate other metrics to evaluate the final value of implicit keywords.

Intra Document Proximity is a metric that states if the terms tend to occur near each other in the document.

$$\square\square(\square,\square) = 1 - \frac{1}{||\square * ||}\square\frac{\square\square\square\square(\square,\square,\square)}{\square\square\square\square h\square\square\square(\square,\square,\square)}$$

D* - set of documents containing both expression A and B
Dist(A,B,d) - the nearest distance in number of words from occurrences of A to B in d
farthest(A,B,d) - the farthest distance from occurrences of A to B in d

The proximity score ranges from 0 to 1. A score closer to 1 indicates that A and B tend to occur very close to each other within the documents where they both appear. If A and B are always right next to each other, IP will be close to 1. Conversely, if A and B are far apart IP score will be reduced.

However, we unfortunately didn't have time to implement the function that calculates IP correctly. Therefore we decided to approximate the value of the Semantic Proximity using only the correlation.

Semantic Proximity combines both intra-document and inter-document proximity to evaluate the overall relatedness of terms across and within documents.

$$\square\square\square\square\square\square\square\square(\square,\square) = \square\square\square\square(\square,\square) \times \sqrt{\square\square(\square,\square)}$$

After calculating all metrics given above, we could go on to calculate the score.

In our specific case the Semantic Proximity has been approximated like this thus:

$$\square\square\square\square\square\square\square\square(\square,\square) = \square\square\square\square(\square,\square)$$

The score of a relevant expression RE for a document d is the sum of its semantic proximities with the top explicit keywords in that document, divided by their rank, in order to give greater importance to those ones that are more relevant for the document.

A Relevant Expression is an implicit keyword of d if it is strongly related with its top explicit keywords. A higher score indicates that RE is strongly associated with the main themes or topics represented by the explicit keywords in the document.

$$\square\square\square\square\square(\square\square,\square) = \square\frac{\square\square\square\square\square\square\square\square(\square\square,\square\square)}{\square}$$

ki -the i-th ranked explicit keyword of d

Having calculated the score for the relevant expressions for each document, we filtered them and we took the 5 with the highest score to obtain the implicit keywords of every document.

11. CONCLUSIONS

In conclusion, while our text mining algorithm successfully identified relevant expressions in the text, there are several areas where future improvements could be made. One significant challenge we encountered was the slow performance of the algorithm, which in our case takes more than one hour. Running the code on large text files resulted in long wait times to obtain results. This is somewhat expected given the size of the data, but it is likely that the code can be optimized to improve its speed and efficiency.

Since we didn't have time to implement IP (Intra-Document Proximity) another future improvement could be the implementation of this one and the addition of this term for the research of implicit keywords. In our project, we limited our approach to approximating semantic proximity only by correlation, without multiplying it by the square root of IP.

Regarding the results, we can conclude that the best performance of the algorithm is with the Dice metric, since the F_metric has a higher value and this means that the Dice metric helps us to find the expressions of the text that are more significant than those found with the other two metrics. We have observed some interesting patterns in the evaluation of our algorithm. For instance, achieving a high recall value has proven to be particularly difficult. This is because it is challenging to find expressions that we consider significant that exactly match those identified by the algorithm. Consequently, the recall values are expected to be low across all metrics used.

Additionally, upon reviewing the relative expressions identified by the algorithm, we noticed that some of them were not particularly meaningful. It was evident that some expressions were unlikely to be considered significant within the context of the text. This observation highlights a potential area for improving the algorithm's accuracy in identifying truly relevant expressions.

Overall, while the algorithm provides a solid foundation for extracting significant expressions from text, there is room for enhancement. Improving the speed of the algorithm and refining its ability to identify meaningful expressions will be crucial steps in future developments. Despite these challenges, the insights gained from this project offer valuable directions for further research and optimization in text mining techniques.