# Sentence Classification and Information Retrieval for Petroleum Engineering

**Thiago F. Ferraz**[1], **Gabriel A. B. A. Ferreira**[1], **Fabio G. Cozman**[1], **Ismael Santos**[2]

[1]Escola Politécnica – Universidade de São Paulo (USP)

[2]PETROBRAS

{thiago.fernandes.ferraz,gabriel.augusto.ferreira,fgcozman}@usp.br

***Abstract.*** *Classifying sentences in industrial, technical or scientific reports can enhance text mining and information retrieval tasks with useful machine-readable metadata. This paper describes a search engine that employs sentence classification so as to search for abstracts from scholarly papers in Petroleum Engineering. The sentences were classified into four classes, based on the popular IMRAD categories. We produced a dataset containing more than 2,200 manually labeled sentences from 278 scholarly articles in the field of Petroleum Engineering in order to be used as training and testing data. The classifier with best results was logistic regression, with an accuracy of 86.4%. The information retrieval system built on top of the classification system yielded a mAP of 0.80.*

## 1. Introduction

Recent decades have seen explosive growth of digital text repositories that contain textual documents of diverse kinds. The development and enhancement of information retrieval systems is extremely important in order to control and manage these repositories [Ladeira and Alvarenga 2012].

Sentence classification can benefit information retrieval, text mining and question answering processes, as it can generate useful machine-readable metadata [Agarwal and Yu 2009, Furtado 2017]. The work reported in this paper focuses on sentence classification within abstracts of scholarly documents (such as scientific papers) in the domain of Petroleum Engineering, with the purpose of generating metadata that can be explored by a search engine. Several authors have proposed machine learning techniques to classify sentences in scientific papers. Many of them are focused on biological or medical fields. For instance, in [Agarwal and Yu 2009] the authors manually labeled sentences from a collection of articles from PUBMED, also adopting the IMRAD classification. Using features such as bag-of-words and presence of citations, they achieved 91.25% accuracy using SVMs. Other proposals can be found that use machine learning methods on sentences in abtracts of scientific papers: Ref. [McKnight and Srinivasan 2003] reports an accuracy of 89,2% and F1-score ranging from 52 to 86%, while [Yamamoto and Takagi 2005] reports an F1-score of 73-89%. The research here reported was designed together with a large oil and gas company, and its results can be of practical use within a key economic sector. In practice, manually producing the metadata to be used by the search engine may be an expensive and time-consuming task. By proposing a system where it can be automatically generated with a certain degree of quality, we both enhance the search task and manage to save valuable resources.

**Table 1. Annotation class (left) and guidelines (right)**

| | |
|---|---|
| Introduction | Background and previous work; not directly linked to present work. |
| Objectives | Tells what the present work is about, i.e. ”in this work”, etc. |
| Methods | Explains the methods used to achieve the Objectives |
| Results / Conclusion | Results obtained by using the referred methods and/or conclusions |

We have explored different machine learning tools, combining distinct classifiers with various feature sets. Sentences are classified into four different categories that reflect the scientific discourse structure: Introduction, Objectives, Methods and Conclusion. These categories were adapted from the commonly used IMRAD scheme (Introduction, Methods, Results and Discussion), to better suit the structure of the abstracts (which usually do not contain discussion). One key contribution of this work is a manually built dataset of labeled sentences extracted from abstracts and used to address the problem of training an automatic classification system on the mentioned categories. This dataset will be made publicly available and may be used by other researchers in the field.

After building a sentence classification system that achieves state-of-art performance results, we use it to feed useful metadata into an information retrieval system. The goal is to allow the user to search the documents more efficiently by letting her provide both the keywords and sentence categories of interest. For instance, one may query “Reduce Costs” on the “Objectives” of the papers. Or “Finite Elements” on “Methods”. We combined language models with word embeddings to consider not only exact matches but also synonyms and related terms to those present on the queries (once initial tests showed that this could enhance the information retrieval performance).

The paper is organized as follows. Section II shows an overview of the whole contribution. Section III discusses the construction of our database and data labelling. Section IV discusses the methods used to perform the classification experiments using supervised learning. Section V presents the results for classification. Section VI and VII show the Information Retrieval methods and results, respectively. Finally, Section VIII presents the conclusions of this work and also discusses possible improvements and future work.

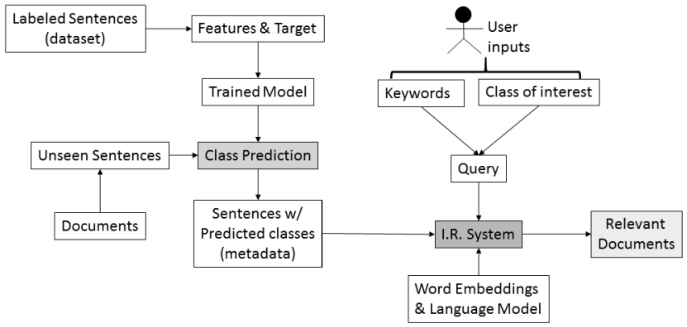


**Figure 1. An overview of the main steps of our pipeline.**

## 2. Overview

Figure 1 illustrates the main steps involved in the development of our search engine. First, we train a machine learning model on a previously labeled dataset (described in Section

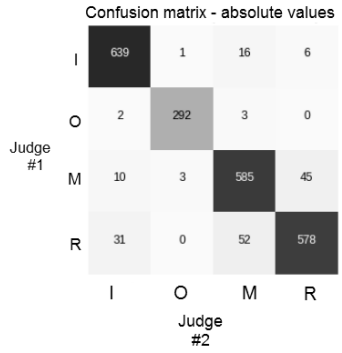


**Figure 2. Confusion matrix showing the agreements/disagreements between the two judges that annotated the dataset. I, O, M and R respectively stand for: Introduction, Objectives, Methods and Results/Conclusion.**

III). This model is then used to predict sentence categories in unseen documents, with the goal of creating relevant information and metadata that can be used by the Information Retrieval system. The documents database is composed of scholarly documents abstracts, in which we treat the sentences as independent units (the classifier assigns each sentence a single class).

As a result of the combination of word embeddings and a language model, an Information Retrieval system allows users to search keywords and also filter the sentences by the categories of interest. Hence one can explore the information that was automatically generated by our sentence classifier (detailed in Sections IV and V). We show that the search engine is enhanced by the generated metadata and also by the usage of word embeddings to smooth language models' estimates (as shown in Sections VI and VII).

## 3. Dataset

In this section we describe the construction of our database and the methods that were used for sentence annotation and quality control. We also discuss the choice of sentence categories and the structure used to organize the data. The dataset is available at http://github.com/thiago-ferraz/Sentence-Classification-and-Information-Retrieval-for-Petroleum-Engineering.

First, we collected 278 papers and its abstracts from the Journal of Petroleum Exploration and Production Technology (from 2011 to 2018), as well as from IEEE Xplore database. All papers were related to Petroleum Engineering. The paper abstracts were then separated into sentences, counting 2420 sentences. Some abstracts were removed from the database because they contained grammar/spelling errors or were poorly written. We thus obtained 2263 sentences. These sentences were independently and manually labeled by the first two of the authors of this paper into the proposed classes (Introduction, Objectives, Methods, Results/Conclusion).

As the sentences were annotated by two judges independently, some guidelines were established so as to reduce the ambiguities that could arise during the process. Table 1 shows the annotation guidance for the different classes. Also, a hierarchy among classes was determined to deal with situations where a single sentence was in more than one category. It was decided that the most important class was Objectives, followed by

**Table 2. Datasets created for the classification experiments. (1, 1) indicates only unigrams and (1, 2) refers to unigrams and bigrams**

| Dataset Number | Positions | PoS-Tags Frequency | Preprocessing of Numerical Strings | N-gram Range |
|---|---|---|---|---|
| 1 | No | No | No | (1, 1) |
| 2 | No | No | No | (1, 2) |
| 3 | Yes | No | No | (1, 1) |
| 4 | Yes | No | No | (1, 2) |
| 5 | Yes | Yes | No | (1, 1) |
| 6 | Yes | Yes | No | (1, 2) |
| 7 | Yes | No | Yes | (1, 1) |
| 8 | Yes | No | Yes | (1, 2) |
| 9 | No | Yes | Yes | (1, 1) |
| 10 | No | Yes | Yes | (1, 2) |
| 11 | Yes | Yes | Yes | (1, 1) |
| 12 | Yes | Yes | Yes | (1, 2) |

Results, Methods and, lastly, Introduction. Most of this methodology was also employed by [Agarwal and Yu 2009].

After labeling, the confusion matrix for this process was computed to show agreement between the judges (Figure 2). In 2094 cases, the judges labeled the sentences the same way, representing 92.5% of all cases. There were 169 sentences (7.5%) with disagreement. These sentences were discarded from the dataset, due to their ambiguities, following usual procedure [Agarwal and Yu 2009].

A popular measure of agreement level between judges is the kappa score (or kappa statistic) [Schütze et al. 2008]. This is a robust indicator of agreement as it compares the proportion of matching cases to the probability of agreement by chance: $\kappa = (p_0 - p_e)/(1 - p_e)$, where $p_0$ stands for the proportion of agreement between judges, and $p_e$ is the hypothetical probability of judges agreeing by chance (which is estimated from the data), as defined by [Schütze et al. 2008]. Kappa score evaluation in our dataset yielded 0.897, a strong level of agreement [McHugh 2012]; hence we can assume that our dataset was reliably labeled.

## 4. Classification Methods

In this section we present the methods used to perform the classification experiments using supervised learning algorithms applied to different dataset configurations.

### 4.1. Features and feature selection

The input set of labeled sentences was randomly divided into 80% for training and 20% for testing with the objective of preventing the overfitting of the classifiers to the data. This division was stratified to preserve the ratios between sentence categories from the original set in the training and testing sets. The best results achieved in the training set were then applied to the test set in order to obtain an empirical estimate of their performance. These empirical results are shown in Section 4.

The features used in the classification experiments were:

1. *Bag-of-words, n-grams and numerical strings*: The base feature set developed was the bag-of-words representation for the text documents. Unigrams (individual words) and the combination of unigrams and bigrams were tested as attributes. The features were extracted after simple preprocessing steps such as the removal of punctuation and the conversion of words to lowercase. We also explored the conversion of every numerical string to a symbol (*#NUMBER#*) ensuring that numbers would be counted as a single feature [Agarwal and Yu 2009].
2. *Position*: This feature captures the position of a given sentence in the abstract, varying linearly from 0 to 1, with "0" indicating that the sentence comes first in the abstract and "1" indicating that it comes last. This feature is important as there is correlation between the position of the sentence and its category. For example, the introduction tends to come before the methods, and these, in turn, before the conclusions and results.
3. *PoS-Tags Frequency*: The NLTK tool [Bird et al. 2009], pretrained on a large corpus, was used in order to get the Part-of-Speech Tags for each word from our dataset. The tags frequency counting was then selected as a new set of features to be considered. For example, one observation (sentence) from our database can have $n$ verbs in the present, $m$ verbs in the past, $p$ adjectives, and so on.

Twelve dataset configurations (with different feature sets), shown in Table 2, were explored in relation to the use of the sentence positions, preprocessed numerical strings, the frequency of PoS-Tags and the n-gram range in the generation of features. All the datasets presented in Table 2 were generated from the same 2094 sentences, but different methods were used to extract features.

The models's hyperparameters were manually adjusted through the application of 5-fold stratified cross-validation in the training set and accuracy evaluation. The same procedure was conducted to select a proper subset of features that maximized the accuracy of each classifier for each dataset configuration.

Reduction of the high data dimentionality can increase the performance of classification tasks [Yun et al. 2007] by decreasing the cost of learning through the removal of noisy/irrelevant data [Tu et al. 2007]. Several metrics can be used to select features, such as Information Gain, Term Frequency, Pearson Chi-square, odd ratio, Gini index, Mutual Information, Document Frequency; we adopted the method proposed by [Elssied et al. 2014] which consists of performing for each continuous predictor $X$ a one-way ANOVA F-test that checks whether all the classes of the target variable $Y$ have the same mean as $X$. For every variable the p-value based on the F statistic is calculated and then used to rank them in descending order.

### 4.2. Classifiers

Three classifiers were chosen to be tested in our classification experiments, based on the state of art for text categorization problems and on particularities of our database: Naive Bayes, SVM and logistic regression. The classifiers predictive performances were evaluated based on the values of accuracy, precision, recall and F1-Score computed through their application on the test set.

Introduced in text classification by [Mosteller and Wallace 1964], the Naive Bayes classifier assumes that features (individual words, n-grams, etc.) are conditionally independent from each other given the class of the observation to which they belong

**Table 3. Feature Selection, Accuracy and F1-Score Results**

| Dataset Number | Naive Bayes | | | SVM | | | Logistic Regression | | |
|---|---|---|---|---|---|---|---|---|---|
| | # feat | Acc | F1 | # feat | Acc | F1 | # feat | Acc | F1 |
| 1 | 404 | 0.699 | 0.699 | 3701 | 0.733 | 0.733 | 3801 | 0.747 | 0.747 |
| 2 | 1351 | 0.718 | 0.718 | 4901 | 0.730 | 0.730 | 2101 | 0.740 | 0.740 |
| 3 | 301 | 0.721 | 0.721 | 4080 | 0.833 | 0.833 | 4320 | 0.847 | 0.847 |
| 4 | 701 | 0.721 | 0.721 | 8801 | 0.831 | 0.831 | 741 | 0.828 | 0.828 |
| 5 | 401 | 0.740 | 0.740 | 3901 | 0.845 | 0.845 | 4500 | 0.864 | 0.864 |
| 6 | 1101 | 0.747 | 0.747 | 8801 | 0.833 | 0.833 | 8000 | 0.847 | 0.847 |
| 7 | 211 | 0.711 | 0.711 | 4110 | 0.835 | 0.835 | 3951 | 0.845 | 0.845 |
| 8 | 661 | 0.721 | 0.721 | 8401 | 0.831 | 0.831 | 8810 | 0.840 | 0.840 |
| 9 | 501 | 0.718 | 0.718 | 3801 | 0.733 | 0.733 | 4001 | 0.752 | 0.752 |
| 10 | 1201 | 0.718 | 0.718 | 6301 | 0.740 | 0.740 | 5301 | 0.740 | 0.740 |
| 11 | 361 | 0.735 | 0.735 | 3451 | 0.847 | 0.847 | 3901 | 0.857 | 0.857 |
| 12 | 1121 | 0.754 | 0.754 | 1141 | 0.838 | 0.838 | 811 | 0.850 | 0.850 |

[Jurafsky and Martin 2018]. Such strong conditional independence assumptions lead the classifier to overestimate evidences for cases in which the data contain highly correlated variables. Even so, empirical studies [Wang and Manning 2012] show that for short textual documents (such as sentences) the Naive Bayes classifier is an effective tool. SVMs (Support Vector Machines) are also often used in text categorization [Wang and Manning 2012]; we have used the linear kernel because of its superior performance in problems where the input data has a large number of dimensions [Joachims 1999]. Finally, logistic regression has some valuable properties. For example, for two perfectly correlated attributes $f_1$ and $f_2$, logistic regression will distribute the weight in the classification process between the terms $w_1$ and $w_2$ (which multiply the variables), while the Naive Bayes classifier will treat them independently by multiplying their probabilities (their weights) in the calculation of the maximum likelihood, overestimating the evidence [Jurafsky and Martin 2018]. Furthermore, logistic regression, as a linear model, is very effective in handling sparse feature spaces, such as the ones generated by the bag-of-words process.

## 5. Classification Results

Table 3 shows, for each classifier, the number of selected features ("# feat") as well as their performance in the test set in terms of accuracy ("Acc") and F1-Score ("F1").

The variation of the number of selected attributes was due to the search for the combination that generated, in each case, the highest accuracy for the classifier in the training set. The Naive Bayes algorithm got its maximum for an average of 693 features, while the Support Vector Machines and logistic regressions required much higher dimensional spaces (averages of 5116 and 4186, respectively).

The importance of feature engineering methods in the text classification task can be seen in several cases presented in Table 3. Comparing the test results in datasets 1 and 3, for example, we can see how the position enhances the score performance. We can observe that its simple addition generated an increase of approximately 13% in the logistic regression accuracy (comparing dataset 1 against 3). The PoS-Tags frequency

**Table 4. Accuracy, precision, recall and F1-Score for the tests performed with datasets 5 and 11.**

| Dataset Number | | Naive Bayes | | | SVM | | | Logistic Regression | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | Prec. | Recall | F1 | Prec. | Recall | F1 | Prec. | Rec. | F1 |
| 5 | Intro | 0.75 | 0.73 | 0.74 | 0.84 | 0.88 | 0.86 | 0.85 | 0.90 | 0.87 |
| | Obj | 0.78 | 0.67 | 0.72 | 0.86 | 0.76 | 0.81 | 0.86 | 0.76 | 0.81 |
| | Methods | 0.75 | 0.78 | 0.76 | 0.84 | 0.84 | 0.84 | 0.88 | 0.85 | 0.87 |
| | Results | 0.70 | 0.74 | 0.72 | 0.85 | 0.86 | 0.85 | 0.87 | 0.89 | 0.88 |
| | Avg | 0.74 | 0.74 | 0.74 | 0.85 | 0.84 | 0.84 | 0.86 | 0.86 | 0.86 |
| | Acc | 0.740 | | | 0.845 | | | 0.864 | | |
| 11 | Intro | 0.74 | 0.73 | 0.74 | 0.85 | 0.87 | 0.86 | 0.86 | 0.88 | 0.87 |
| | Obj | 0.79 | 0.66 | 0.72 | 0.86 | 0.76 | 0.81 | 0.88 | 0.76 | 0.81 |
| | Methods | 0.73 | 0.78 | 0.75 | 0.83 | 0.84 | 0.83 | 0.86 | 0.85 | 0.86 |
| | Results | 0.71 | 0.74 | 0.73 | 0.85 | 0.88 | 0.86 | 0.84 | 0.8 | 0.88 |
| | Avg | 0.74 | 0.74 | 0.73 | 0.85 | 0.85 | 0.86 | 0.86 | 0.86 | 0.86 |
| | Acc | 0.735 | | | 0.847 | | | 0.857 | | |

have also shown a great discriminative power as it can be seen in the results related to datasets 3 (without those features) and 5 (containing those features). The Bigrams frequency increased the accuracy for every test performed with the Naive Bayes classifier, but had an opposite effect on the SVMs and logistic regression. Finally, the preprocessing of numerical strings generated a mixed effect over the classifiers performance, sometimes increasing their accuracy (tests performed with datasets 6 and 12) and sometimes decreasing it (Naive Bayes and logistic regression tests in datasets 3 and 7).

It can be noted that logistic regression always outperformed the other classifiers except for datasets 4 and 10. Logistic regression also obtained the best overall results of 85.7% and 86.4% in terms of accuracy, for the tests performed with the datasets 11 and 5, respectively (as described by Table 2). For those cases (datasets 11 and 5), which had the best score results, a more complete report is presented in Table 4. There, the precision, recall and F1-Score for each class (Introduction, Objectives, Methods and Results) is presented together with their average values.

In Table 5 we show some examples of sentences classified by our best model (logistic regression trained on dataset 5). The table carries two sentence examples per class: a correctly classified sentence and a misclassified one.

## 6. Information Retrieval Methods

In this section we describe the integration between the classification and the information retrieval tasks. The latter benefits from being fed with metadata generated by the former.

The simplest IR strategy is Boolean retrieval, where queries are words combined and connected with operators AND, OR and NOT. This method treats each document as a set of terms; as the queries are Boolean, the results are either TRUE or FALSE, which means a document is either relevant or not. There is no way of determining a scale or scoring for relevance, or even a way of sorting the positive outputs.

To assign scores to each document given a certain query, one could use a probabilistic language model, such as a unigram model. In such a model, we estimate the probability of a document given a query, using the frequency of the terms present in both the query and the document. The probability of a document given a query can be expressed as:

$$P(D|q) = \frac{P(q|D) \cdot P(D)}{P(q)}, \tag{1}$$

where $D$ stands for Document and $q$ stands for query. According to [Zhai and Lafferty 2004], (1) can be simplified to make $P(D|q) \propto P(q|D)$. The probability of a query (consisting of words $w_1, \ldots, w_n$) given the document can be then expressed as:

$$P(q|D) = \prod_{w_i \in q} P(w_i|D). \tag{2}$$

When dealing with a unigram model, each word's frequency is considered independently of their order in the document. Hence, we can estimate $P(w|D)$ as:

$$\hat{P}(w|D) = tf(w, D) = N(w, D)/||D||, \tag{3}$$

where $tf$ indicates the term frequency function, $N(w, D)$ is the number of times a certain word $w$ appears in document $D$ and $||D||$ represents the total number of words of a certain document [Zhai and Lafferty 2004]. The main difficulty with this kind of model happens when certain word $w$, present in the query, is missing in a document. In this case, the probability in (3) will be zero and so will be the probability of the query, as seen in (2). To address this problem, several authors propose smoothing methods that avoid the assignment of zero probability to the estimate $\hat{P}(w|D)$ [Chen and Goodman. 1998].

As previously discussed, one of our goals was to build an IR system that considers not only perfectly matching words but is able to deal with synonyms and similar words. Accordingly, we used a word embedding representation in order to represent the words as vectors and then be able to define and compute distances between terms. We used a pre-trained embedding model called "Common Crawl" [Pennington et al. 2014]. It was trained on Wikipedia Articles and more than 42 billion tokens were used, resulting in a 1.9M words vocabulary.

Our approach for IR consists in using a word embedding model to smooth the language model estimates. [Ganguly et al. 2015] defines the probability of transforming a term $t'$ into a term $t$, given a document:

$$\hat{P}(t \mid t', D) = \frac{sim(t, t')}{\sum_{t'' \in D} sim(t, t'')}, \tag{4}$$

where $sim(t, t')$ represents the Cosine Similarity. By refining Expression (4), we propose a smoothing expression that takes into account the most similar word in the document, regarding each term in the query. Intuitively, it considers not only exact matches between what is being searched and what is in the documents, but also synonyms or related words:

$$\hat{P}_\beta(w|D) = (1 - \beta)\, tf(w, D) \;+\; \beta \frac{1}{\sum_{w' \in D} sim(w', w)} \max_{t' \in D}[sim(w, t')]. \tag{5}$$

**Table 5. Examples of sentences from test set, with their respective ground-truth label and predicted label (by our best model - logistic regression trained on dataset 5).**

| Sentence | Ground-truth | Predicted |
|---|---|---|
| "In the Tahe oilfield in China, heavy oil is commonly lifted using the light oil blending technology." | Introduction | Introduction |
| "Xujiahe Formation is a set of terrestrial clastic rocks with low compositional maturity, low cement content, and medium textural maturity." | Introduction | Methods |
| "This paper investigates the potential of high acyl GLG as additive for drilling mud." | Objectives | Objectives |
| "The present work sought to determine whether or not a commercially available simulator could accurately simulate results from core flooding experiments." | Objectives | Results |
| "These flow rates were used to characterize the performance of the jet pump." | Methods | Methods |
| "The properties are controlled at such values that the mud provides optimum performance." | Methods | Introduction |
| "Furthermore, it has been concluded that aquifer strength has a little effect on coning behavior during oil production process" | Results | Results |
| "Furthermore, simulations using the X model suggest an incremental oil recovery factor of 11% OOIP due to surfactant-polymer flooding." | Results | Methods |

In Expression (5), $\beta$ is a parameter that controls how much weight is given to the occurrence of related words versus the exact matches. The expression is divided in two complementary terms: the first one, multiplied by $(1 - \beta)$, deals simply with the term frequency $(tf)$, as defined in Expression (3); and the second one, multiplied by $\beta$, deals with the occurrence of synonyms and related words. $\beta$ parameter must be tuned for each specific application, as it happens in most smoothing techniques.

We integrate our smoothed language model with the sentence categorization task by considering the probabilities of each sentence (document) belonging to the desired class $\gamma$: $P(\gamma|D)$. Hence, we define the score of a document given a query $q$ and a desired class $\gamma$ as:

$$\begin{aligned} score(D, q, \gamma) &= P(q, \gamma|D) = P(q|D) \cdot P(\gamma|D) \\ &= \Big[ \prod_{w \in q} \hat{P}_\beta(w|D) \Big] \cdot P(\gamma|D). \end{aligned} \quad (6)$$

## 7. Information Retrieval Results

In our tests, the score for a particular article was then given by its sentence with the highest score. In other words, the most relevant sentence dictates the score of the entire abstract.

To validate this method, a new dataset of 320 scientific papers, different from the ones used in the training of the sentences classifier, was collected from the Journal of

Petroleum Exploration and Production Technology.

The importance of metrics such as precision and recall in the evaluation of IR systems is related to the type of application. Some applications such as web searches tend to favor precision because the user is much more interested in having relevant results on the first page than obtaining all possible relevant results. On the other hand, on tax or legal search systems, which involve more detailed tasks, sensitivity plays a more important role because it may not be tolerable to lose certain results [Manning et al. 2008]. We do not focus here on sensitivity, since our main concern is to evaluate the returned results relevance rather than checking whether all useful results are found. In fact, in search systems like ours, the larger the database, the lower the importance of the recall.

A set of 50 queries was evaluated (following guidelines in the literature [Manning et al. 2008]). Queries were independently suggested by volunteers after they quickly inspected the topics in the database. They were instructed to provide queries consisting of keywords, with one to three words each, and to indicate, for each query, one or more classes of sentences in which they were interested (Introduction, Objectives, Methods or Results). Results were evaluated by the first two authors, who, independently, indicated, for each result, whether it was relevant or not according to the corresponding query. The results were only considered relevant when marked this way by both judges.

The analysis of the set of 50 queries yielded a mAP score of 0.80 with a standard deviation of 17.1%. The results retrieved first papers whose abstracts contained all the words from the query and, after that, documents containing some of the query words as well as related words (such as synonyms), indicating success of our algorithm. An example of result is the following. Given the query (keywords: ”Reduce Costs”, class: ”Objectives”), the system returned documents containing: “reduce costs”, “cost reduction” and “eliminate expenditures”.

## 8. Conclusion

Our classification results match or exceed the results of similar efforts in the literature [Agarwal and Yu 2009, McKnight and Srinivasan 2003, Yamamoto and Takagi 2005]. Our system was able to classify sentences from Petroleum Engineering abstracts into the main categories of the scientific discourse, with an accuracy of 86% (comparable to the best in the literature) and an F1-Score of 0.81-0.88 (higher than related work). We should note that one of our contributions is a dataset that will be made publicly available. Most of the literature focuses on biological and medical texts, while here we have dealt with a literature of huge importance, namely the literature related to the oil industry. Although we used scientific texts, our techniques can be successfully adapted to other industries. To this end, two main aspects can contribute: 1) the level of vocabulary similarity between technical reports and scientific papers' excerpts; and 2) the conciseness typically present on both abstracts and technical reports.

The results of the tests carried out on the IR system show, both qualitative and quantitatively, that the desired result was reached: queries return on average 80% of relevant results and, in addition, it is possible to find documents that contain similar words to the ones in the query, but are not exact matches. In addition, filtering the automatically categorized sentences has also proved to be a promising tool.

Future work that could improve the classification system includes increasing the

number of labeled observations in the training set, especially the Objectives class, aiming at improving the classification accuracy and the recall of this class that presented lower values in comparison to the others (0.76 recall for the best classifier). The annotation process could also be improved by reducing its bias. This could be done by having multiple independent researchers labeling the sentences. Testing other classification methods could also help to increase system accuracy. Regarding the IR system, some improvements could be done in future work: 1) test different word embedding models (or even train one on scholarly papers from a certain specific field); 2) test the proposed algorithm on a more statistically significant sample (i.e. test on more examples); 3) reduce the bias of the tests by having more judges. Another important step would be to actually test our entire pipeline on documents of a distinct nature: reports generated by the industry, technical texts and guides, and so on. It would be important to validate the presented ideas and provide helpful insights on how to adapt the models to other contexts of the real world.

## Acknowledgements

The author Fabio G. Cozman has been partially supported by the Conselho Nacional de Desenvolvimento Cientifico e Tecnologico (CNPq), grant 312180/2018-7. The work was also supported by the Fundação de Amparo a Pesquisa do Estado de São Paulo (FAPESP), grant 2016/18841-0, and also by the Coordenação de Aperfeiçoamento de Pessoal de Nivel Superior (CAPES) - finance code 001.

## References

Agarwal, S. and Yu, H. (2009). Automatically classifying sentences in full-text biomedical articles into introduction, methods, results and discussion. *Bioinformatics*, 25(23):3174–3180.

Bird, S., Klein, E., and Lopen, E. (2009). *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. O'Reilly Media, Inc.

Chen, S. F. and Goodman., J. (1998). An empirical study of smoothing techniques for language modeling. *Harvard Computer Science Group Technical Report*.

Elssied, N. O. F., Ibrahim, O., and Osman, A. H. (2014). A novel feature selection based on one-way anova f-test for e-mail spam classification. *Research Journal of Applied Sciences, Engineering and Technology*, 7(3):625–638.

Furtado, P. H. T. (2017). Interpretação automática de relatórios de operação de equipamentos. Master's thesis, Pontifícia Universidade Católica do Rio de Janeiro.

Ganguly, D., Roy, D., Mitra, M., and Jones, G. J. (2015). Word embedding based generalized language model for information retrieval. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, pages 795–798, New York, NY, USA. ACM.

Joachims, T. (1999). Transductive inference for text classification using support vector machines. In *ICML*, volume 99, pages 200–209.

Jurafsky, D. and Martin, J. H. (2018). *Speech and Language Processing*, volume 3. Pearson London.

Ladeira, A. P. and Alvarenga, L. (2012). Processamento de linguagem natural: em busca de evidências temáticas nas publicações nacionais e contemporâneas.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

McHugh, M. L. (2012). Interrater reliability: the kappa statistic. *Biochemia medica: Biochemia medica*, 22(3):276–282.

McKnight, L. and Srinivasan, P. (2003). Categorization of sentence types in medical abstracts. In *AMIA Annual Symposium Proceedings*, volume 2003, page 440. American Medical Informatics Association.

Mosteller, F. and Wallace, D. (1964). Inference and disputed authorship: The federalist.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543.

Schütze, H., Manning, C. D., and Raghavan, P. (2008). *Introduction to information retrieval*, volume 39. Cambridge University Press.

Tu, C.-J., Chuang, L.-Y., Chang, J.-Y., and Yang, C.-H. (2007). Feature selection using pso-svm. *IAENG International Journal of Computer Science*, 33(1):111–116.

Wang, S. and Manning, C. D. (2012). Baselines and bigrams: Simple, good sentiment and topic classification. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers-Volume 2*, pages 90–94. Association for Computational Linguistics.

Yamamoto, Y. and Takagi, T. (2005). A sentence classification system for multi biomedical literature summarization. In *Data Engineering Workshops, 2005. 21st International Conference on*, pages 1163–1163. IEEE.

Yun, C., Shin, D., Jo, H., Yang, J., and Kim, S. (2007). An experimental study on feature subset selection methods. In *7th IEEE International Conference on Computer and Information Technology (CIT 2007)*, pages 77–82. IEEE.

Zhai, C. and Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214.