International Journal of Business Intelligence Research
Volume 11 • Issue 2 • July-December 2020

Text Mining Business Policy Documents:
Applied Data Science in Finance

Marco Spruit, Utrecht University, The Netherlands
https://orcid.org/0000-0002-9237-221X

Drilon Ferati, Utrecht University, The Netherlands

ABSTRACT

In a time when the employment of natural language processing techniques in domains such as biomedicine, national security, finance, and law is flourishing, this study takes a deep look at its application in policy documents. Besides providing an overview of the current state of the literature that treats these concepts, the authors implement a set of natural language processing techniques on internal bank policies. The implementation of these techniques, together with the results that derive from the experiments and expert evaluation, introduce a meta-algorithmic modelling framework for processing internal business policies. This framework relies on three natural language processing techniques, namely information extraction, automatic summarization, and automatic keyword extraction. For the reference extraction and keyword extraction tasks, the authors calculated precision, recall, and F-scores. For the former, the researchers obtained 0.99, 0.84, and 0.89; for the latter, this research obtained 0.79, 0.87, and 0.83, respectively. Finally, the summary extraction approach was positively evaluated using a qualitative assessment.

KEYWORDS

Applied Data Science, Automatic Summarization, Financial Industry, Information Extraction, Keyword Extraction, Natural Language Processing, Policy Documents

1. INTRODUCTION

Data are the pollution of the information age, since they are created and are here to stay (Spence, 2010). This increase in the flow of data that organizations create and collect, necessitates the need to leverage these resources and extract information and subsequent knowledge. The large stream of data unveiled two data formats, namely structured and unstructured data, with each of them requesting different treatment methodologies to derive knowledge. Although many argue that this process is less-time consuming when the data have a consistent representation and a predefined structure, only 20% of the data that organizations have, are actually found in this manner. The rest of the data are found in an unstructured format (Grimes, 2008). These data have no consistency in appearance and are usually text-heavy, making it a challenge to extract patterns and relationships from and among

DOI: 10.4018/IJBIR.20200701.oa1

them. This has called for the introduction of Text Mining (TM) as a discipline that "analyses text to extract information that is useful for particular purpose" (Witten, 2004). We consider TM to be a multidisciplinary field which utlizes data mining techniques, information extraction, information retrieval, machine learning, natural language processing (NLP), statistical data analysis and graph theory, among others (Miner et al., 2012). The wide range of techniques that TM fosters, together with its applicability in domains such as biomedicine, national security, finance, social studies, law and so on, show the prominence of such data analyzing techniques (Bholat, Hansen, Santos, & Schonhardt-Bailey, 2015; Fan, Wallace, Rich, & Zhang, 2006; Friedman, Johnson, Forman, & Starren, 1995; Haug, Ranum, & Frederick, 1990; Menger, Scheepers, & Spruit, 2018; Zhao, 2013).

Nevertheless, not all disciplines have been able to taste the same riches. Policies are industry-wide documents that represent written guidelines of acceptable actions to which organizations must adhere. Financial institutions, especially banks, can be thought of industries that have a high number of policies in place. These organizations continuously introduce policies, in order to be fully complaint with regulations that governing bodies impose. Nevertheless, even though such documents are found across industries, they still lack in standardization (A. I. Anton & Earp, 2003) and their domain-specific language often makes them incomprehensible (A. Anton & Earp, 2004). Considering the importance of these documents for the business, but at the same time their inconsistent and exhausting representation, a perception was first created in (Spruit & Ferati, 2019) that TM and its techniques can be used to bring order and understanding into them. This is what motivated the compilation of the following three related research questions (RQ):

1. To what extent has TM been applied on policy documents?
2. Which TM techniques or frameworks have been applied on policy documents?
3. Which TM techniques can be used to obtain information that would enable an easier navigation through the policies?

Nevertheless, in our attempt to validate this perception, the scientific body of literature showed a landscape different from what was anticipated. Thus, next to providing an overview of the current state of literature that treats the concept of using TM on internal bank policies, this paper also introduces a novel TM framework for processing internal business policies. From a design science research (DSR) perspective, we present a meta-algorithmic model (MAM) as the main DSR artifact to structure and support analyses of internal business policies, which was validated through a case study at one of the biggest banks in The Netherlands (Spruit & Jagesar, 2016). The rest of the paper is structured as follows. In Section 2 we present the research approach, which is followed by the related literature in Section 3. Section 4 presents the case study methodology, whereas the results of the case study and implications are presented in Section 5. Conclusions are formulated in Section 6, whereas discussions, limitations and future work are presented in Section 7.

2. RESEARCH APPROACH

An answer to our research questions calls for an extensive review of the literature that explores the use of TM on policy documents. We employ the snowballing approach as defined in (Wohlin, 2014) to determine to what extent these concepts have been addressed before. Such a review, besides providing empirical evidence of the state of literature, also provides the necessary knowledge foundations to guide the further development of the MAM artefact. The method development is in line with the Applied Data Science approach outlined in (M. Spruit & Lytras, 2018), which integrates the Design Science Research (DSR) approach as introduced by (Hevner, March, Park, Ram, & Ram, 2004) and its guidelines for acquiring a better understanding of the requirements which subsequently leads to

an effective execution of the research. In short, when these guidelines are applied to the study at hand, they are as follows:

1. **Problem Identification:** The extensive amount of internal bank policies, together with their text-heavy format, makes it challenging to read and extract information from them;
2. **Definition of the Objectives of a Solution:** We devise a new method that will enable the automatic analysis of such documents and extract valuable information from them;
3. **Design and Development:** The method will rely on the combined use of TM and Natural Language Processing (NLP) techniques, which will be derived from the literature;
4. **Evaluation:** The use of statistical methods together with expert opinions will be employed to evaluate the framework;
5. **Communication:** The validity and generalizability of the devised framework will be communicated based on the results.

The validation of the framework, which is the Meta-Algorithmic Model artifact, takes place through a case study at one of the most prominent international banks in The Netherlands. Considering that this experiment utilizes domain-specific documents, a particular set of domain knowledge is needed to be able to meaningfully evaluate the outcome. Since such domain knowledge is hardly found among engineers, the evaluation of the algorithmic outcome was performed with a form of acceptance testing, by experts that reside within the organization that facilitated this study. Acceptance testing is an entity of Black-Box testing (Copeland, 2003) where the evaluators provide comments regarding the results that have derived from the artefact. Their comments will indicate how relevant a result is with respect to a policy document.

3. RELATED WORK

The extensive amount of unstructured data has driven researchers to make use and leverage techniques from the TM discipline. An indication of this is the available literature, which is quite rich with publications that examine the use of TM and its techniques in a variety of disciplines. Nevertheless, to our surprise, only a relatively small number of publications treated the use of TM on policy documents. Furthermore, most of the policies used in these studies were privacy policies, which are more commonly known as *"Terms and Conditions"* or *"Terms of Use"*. Thus, driven by the ambiguity and text-heavy format of privacy policies, (Fei Liu, Ramanath, Sadeh, & Smith, 2014a) investigated methods that would bring some understanding into them. Given that most privacy policies address similar issues, such as the collection of a user's location, contact details, financial information and so on, the authors present a classification method that aligns sections of different policies, based on their thematic similarity. In their investigation, the authors used policies from the top-visited websites. Upon splitting the policies into sections and paragraphs, the authors tried two classification approaches, namely the Hidden Markov Model (HMM) and the clustering algorithm of (Zhong & Ghosh, 2005). The research indicated that at the section level the clustering algorithm performed better, whereas at the paragraph level HMM was more appropriate.

A similar investigation on website privacy policies was conducted by (Massey, Eisenstein, Antón, & Swire, 2013) on a corpus of 2,061 privacy policies from Google's Top 1000 visited websites and Forbes 500 companies. Their research focuses on three characteristics of privacy policies. First, they assess the readability of these documents for requirement engineering. Second, they examine if automated TM can indicate whether a policy contains requirements outlined as either privacy protection or vulnerability. Third, they determine whether the identification of privacy protection and vulnerability can be generalized to other policies. The readability aspect of the policies was evaluated with the use of readability metrics such as: Flesch Reading Ease (FRE), Flesch Grade Level (FGA) (Flesch, 1948), Automated Readability Index (ARI) (Count, Reading, & Personnel, 1975), SMOG

(McLaughlin, 1969) and FOG (Gunning, 1952). The results of this evaluation shows that policies are indeed difficult to read. Whereas, for indicating whether the policy represents a privacy protection or vulnerability and whether it can be generalized to other policies, the authors use Topic Modeling (Blei, 2012; Syed, Borit, & Spruit, 2018). Topic Modeling is able to reveal hidden themes within a large collection of documents based on three assumptions. First, the documents are made of topics and the topics are made of words. Second, topic identification is done automatically rather than manually. Third, the topics are shared across the collection of documents. The results from this experiment show that Topic Modeling can indeed be used in determining whether a policy holds requirements that are expressed either as privacy protection or vulnerability.

Given that websites are required by law to disclose the data that they collect, share and store, (Costante, den Hartog, & Petkovic, 2012) use TM and machine learning techniques to associate privacy categories with policies that address these categories. The authors propose a system that uses these techniques to assess the completeness of the privacy policy. Text categorization is used to label the paragraph to the thematic categories it belongs, whereas, machine learning is used to build an automatic classifier by learning from a pre-classified set of documents. In their attempt, the authors investigated multiple classification algorithms, such as Naïve Bayes, Linear Support Vector Machine (LSVM), Ridge Regression, k-nearest neighbor (k-NN), Decision Tree (DT), and Support Vector Machine (SVM). LSVM and Ridge Regression turned out to be the most successful techniques in this case.

On the other hand, (Stamey & Rossi, 2009) introduce Hermes in their study. Hermes is a system that provides a better understanding of privacy policies by determining whether the policy discloses what user information is being collected, what technology is used to collect these information and with whom may the collected user information be shared. It does so by considering policies as a collection of topics, with each topic having its own representative words. The results of their experiment show that by using Latent Semantic Analysis (LSA), the system is able to identify the main topics of the policy and the most significant words for each topic together with a collection of words that express ambiguity. On top of this, Hermes also provides a score that shows the similarity of the user entered policy with a typical privacy policy.

One of the sparse publications that examines internal business policies, as the subject of their investigation, is the research of (Li, Wang, Zhang, & Zhao, 2010). In their work, the authors turn their attention towards policies that define or constrain business processes, or as they are commonly known as process policies (*i.e.* order fulfillment policy, travel reimbursement policy, product development policy etc.). By employing TM and IE techniques, the authors propose a policy-based process mining (PBPM) framework that automatically extracts process models from business policies. This framework defines four main steps. First, process policies are separated from non-process policies. Second, major process components such as tasks, organizational resources and data items are identified with the use of Named Entity Recognition (NER). Third, the relationship among the identified entities is extracted. Fourth, process models are constructed based on the identified components.

(Costante, Sun, Petković, & den Hartog, 2013) advocate that from all the uncertainties that come with reading a privacy policy, the users should be well aware of what data is being collected and thus judging whether that is acceptable. Thus, the authors provide a solution that analyses policies automatically and identifies which user details are being collected by the provider. In order to extract the list of data that is being collected, the authors utilize Information Extraction (IE). This experiment managed to achieve a 80% accuracy in extracting relevant information, which the authors judge to be reasonably high.

(Xiao, Paradkar, & Xie, 2011) seek to extract Access Control Policies (ACP) from requirements documents. These policies are used to specify to what resources specific users have access to, and as such, they represent a mechanism that prevents security vulnerabilities. In an attempt to avoid the manual extraction of these specifications, the authors propose a three-step NLP technique for automatically extracting these instances. First, the text is analyzed linguistically, with words and

phrases being annotated based on their semantic meanings. Second, it constructs model instances with the use of annotated words and phrases. And third, it transforms these model instances into a formal specification. The authors translate this approach into a new system called Text2Policy. The evaluation of this system showed that Text2Policy performs reasonably well and its automation has a significant impact on reducing the effort for extracting security policies from software requirement documents. Furthermore, (Brodie, Karat, & Karat, 2006) introduce SPARCLE (Service Privacy ARchitecture and CapabiLity Enablement) as a system that will assist organizations in linking privacy policies with their implementation. Here, the authors employ a similar approach as in the work of (Xiao et al., 2011) which has resulted to achieve a 94% accuracy in parsing the policies.

(Ammar, Wilson, Sadeh, & Smith, 2012) also investigates the possibilities of automatically categorizing privacy policies. By analyzing a corpus of 56 privacy policies, the authors found three main concepts that were shared through the policies. Those concepts are related to the "ability to leave the services", "transparency on law enforcement" and "notify before changing the terms". Thus, depending on whether the identified concepts appeared in the policies, the authors used logistic regression to classify them into two categorical labels, namely: "present" and "absent". The report on their findings showed that the approach was to some extent suitable for identifying the "transparency on law enforcement" concept, and not that suitable for the "ability to leave the service". There was no trace of reporting with respect to the third concept. Further works describe HMM and clustering as NLP techniques to automatically segment policies based on the privacy issues they address (Fei Liu, Ramanath, Sadeh, & Smith, 2014b; Ramanath, Liu, Sadeh, & Smith, 2014).

(Michael, Ong, & Rowe, 2001) poses the need for organizations to have a "policy workbench". This integrated system would enable users to access, search and update policies from a centralized repository, through an interface, with the sole purpose of facilitating policy adherence. Different from the other studies, this research does not focus on privacy policies, instead they distinguish three types of business policies: meta-policy, goal-oriented policy, and operational policy. Their approach towards developing the "policy workbench" was based on the architecture that (Sibley, Michael, & Wexelblat, 1991) describes. This is an extensible architecture that theoretically describes how such a system should be developed and what the main entities are of the architecture. Nevertheless, for this study the authors only focused on the first entity, that of processing the user input, or as they call it Natural Language Input Processing (NLIP). The system expects a user input that is formulated in a natural human language. It then automatically, tokenizes, parses (LT CHUNK) and tags (LT POS) the inputted text, in order to identify and extract main elements (*i.e.* subject, object, attributes and verbs) (Grover, Matheson, Mikheev, & Moens, 2000), however, with mixed results. Since then related work in legal texts summarization has shown more promising results in single-document summarization, as recently noted in (Kanapala, Pal, & Pamula, 2019).

Furthermore, (Sadeh et al., 2013) investigate ways to improve the readability of privacy policies with the use of NLP, privacy preference modelling, crowdsourcing and policy interfaces. With regard to the language processing aspect of the policies, the authors suggest that in analyzing privacy policies researchers should go beyond text categorization. Nevertheless, only a conceptual framework is presented in this publication, with no proven results. As far as NLP goes, the authors suggest that they will experiment with text categorization and semantic parsing, but do not specify the required linguistic and statistical techniques for these tasks.

To conclude, a rather different yet interesting type of investigation that addresses financial policies, is the work of (A. I. Anton & Earp, 2003). In their work, the authors rely on goal mining heuristics to extract relevant information, and not on TM techniques.

4. CASE STUDY METHODOLOGY

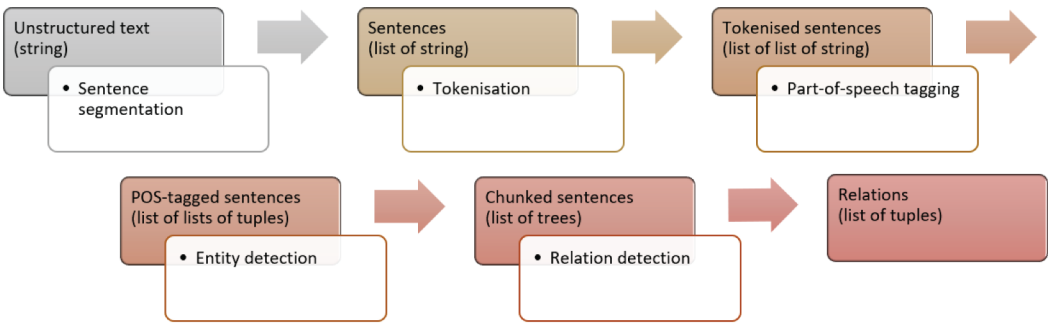
We examine the following three TM techniques in this study: information extraction, automatic summarization and keyword extraction. Given that the literature did not provide much evidence on the

use of specific TM techniques for internal business policies, the mere use of these techniques on such documents can already be considered a novelty. Moreover, the selection of the techniques was also in accordance with the requirements that the facilitator of the study expressed. Within their content, policies often refer to other internal policies, for various purposes. These references can be found dispersed throughout the whole content of policies which can consist of numerous pages. Stakeholders find this information important, indicating that the automatic extraction of such information would be beneficial. Additionally, for each document they requested a list of most descriptive words, which can later be used to tag and index these documents in a centralized repository. Next to this, having a concise summary of each policy would enable the stakeholder to get a timelier understanding of the context of the document. To endorse this study, the stakeholders provided a corpus of 25 internal policies, from which two policies were unusable, since one of them was written in the Dutch language and the other turned out to be nonexistent. This resulted in a final corpus of 23 internal bank policies. Upon investigating the corpus of documents, some data preparation steps took place. Initially all policies were converted from PDF format to plain text format (.TXT). This transformation was done with the help of a script written in Python. Here, Python was selected, given that it is a high-level general-purpose programming language, with a large bag of standard libraries that enable it to be used efficiently in scientific computations (Sanner & others, 1999). Next to this, some data cleaning took place, by removing the cover page, table of content, tables and appendices. Since the investigation concerned only the main content of the document, these steps were taken as a preventive measure so that they will not skew the data. Furthermore, Python was also used in constructing the algorithm for processing the textual data. This was based on the decision to use the Natural Language Toolkit (NLTK) package, which is one the most popular Python libraries specialized in processing textual data (Bird, Klein, & Loper, 2009).

4.1. Reference Extraction

In employing IE one can choose to rely on linguistic processing of the text or on a keyword matching approach, to extract relevant information from the document. In this research we rely on the linguistic processing of text approach, more precisely we rely on the IE architecture that NLTK supports (Figure 1). This process was initiated with the splitting of the raw text into

Figure 1. Information extraction architecture



sentences. This was made possible by the sentence segmenting module that NLTK provides. Furthermore, the split sentences were further tokenized into respective words, which was done by using the NLTK word tokenizer module. The next step in the pipeline was to tag the words based on their semantic and syntactic structure. Such a step assigns a Part-of-Speech (POS) tag to the words based on their role in the textual content (Voutilainen, 2003). Due to the lack of a

domain-specific tagger, we use the default NLTK POS tagger, which utilizes the Penn Treebank (Marcus, Santorini, & Marcinkiewicz, 1993) annotation corpus. This step transforms the token representation of the words into a (word, tag) tuple representation. Having such a representation enables us to investigate the tagged corpus and identify the representation of the relevant entities. The significance of this step is that it enables to distinguish the signal from the noise. Thus, by investigating the corpus, we distinguish the relevant entities and construct a chunking grammar based on their POS representation. We construct this chunking grammar with the help of Regular Expressions (RegEx), which are subsequently parsed on the annotated text. When parsed on the textual content, the chunker identifies entities that we deemed as relevant, and groups them together in a single tree representation.

4.2. Automatic Summarization

Automatic summarization (Kanapala et al., 2019; Larson, 2012) defines two groups of summarizing algorithms, namely extractive and abstractive algorithms. These algorithms differ in the approach that they employ for constructing summary representations. Extractive algorithms construct the summaries by using the most important sentences of the textual document and concatenating them into a consisted summary. Contrarily, abstractive summaries may not always draw on the same concepts as the ones that the original text contains. It usually reuses the main phrases of the document and constructs them in a manner that would convey the message. In addition to this, these two algorithms are further categorized based on their appliance, which can be either a multi-document or single-document summarization algorithm. As the name implies, a multi-document summarization algorithm generates a single summary from the entire corpus of documents, whereas a single-document summarization algorithm generates an individual summary for each document. These factors have scoped our algorithm selection options to an extractive, single-document summarization algorithm. This decision, in turn, has led us to use the TextRank approach.

TextRank was introduced in (Mihalcea & Tarau, 2004) and it has its roots embedded in the PageRank algorithm of (Page & Brin, 1997), utilizing the same logic. As such, TextRank is a graph-based approach, that uses the knowledge drawn from the entire text to construct a graph representation, on which graph the PageRank formula is applied to determine the most important vertices. Adhering to the same methodology, the content of each policy is first split into sentences. Similar to IE, this is done with the NLTK sentence segmenter. This segregation enables to construct a graph representation of the policy where each individual sentence represents a vertex (node) in the graph. To add an edge between vertices, the algorithm relies on the "recommendation" concept. This concept is built under the assumption that a given sentence recommends another sentence to read, based on their resemblance. TextRank is still considered highly effective, and even in recent years several improvements over the original TextRank have been proposed, *e.g.* to better mine users' interests expressed in microblogs and to improve multi-document summarization in online argumentation (Niu & Shen, 2019; Xiong, Li, Li, & Liu, 2018).

To compute the similarity between vertices, the Levenshtein Distance (LD) was employed (Levenshtein, 1966). Such a metric has been used to measure the error rate in text entry (Soukoreff & MacKenzie, 2001), measure the syntactic variation of different dialects (Heeringa, Nerbonne, Van Bezooijen, & Spruit, 2007; M. R. Spruit, 2006), extract keywords (Renz, Ficzay, & Hitzler, 2003), and to extract features of graph representation (Wilson & Hancock, 2004). For two distinct vertices in the graph (V_1 and V_2), LD determines the insertion, deletion and update that V_1 needs, to become same as V_2, assigning this score as an edge between the vertices. Such a score was computed for all the vertices in the graph, adding an edge between them. With this in place the PageRank equation is computed on the graph, which determines the importance of each vertex in the graph, enabling us to retrieve a summary with the highest scoring vertices.

4.3. Keyword Extraction

Another reason that makes TextRank appropriate for this study, is its ability to extract keywords as well. When extracting keywords, the analysis took place on a lower level, more precisely on a word level. Thus following the same logic as in automatic summarization, we further split the sentences into individual words with the use of the NLTK word tokenizer. These tokenized words represent the vertices of the graph. Nevertheless, these changes have an impact on the size of the graph which increases exponentially given the large number of vertices. Thus, in order to reduce the density of the graph, three word filtering methods are applied. First, all "stop words" are removed. These are frequently appearing words that assist in creating an idea, when used in a sentence, but do not represent a significant meaning in themselves (Rajaraman & Ullman, 2011) (*i.e.* the, and, or, that, this). Next, a syntactic filter is applied. This filter determines valuable and invaluable words based on their POS representation. Similar as with IE, the words are first tokenized and annotated with their POS tag. From here, all the tokens with an invaluable syntactic representation are filtered out. The third filter is designed to remove duplicate values from the graph, as advocated by (Mihalcea & Tarau, 2004). Then, the similarity between vertices is computed using LD, thus creating the edges between them. Subsequently, the PageRank formula is applied to the graph to determine the most important vertices. When extracting a list of references, the authors advocate that 1/3 of the content should be retrieved as potential keywords. Nevertheless, even with all the filtering, the graph still contains several thousand vertices. Thus, adhering to such a ratio may still result in an exhausting list of potential keywords. Therefore, we only consider the Top 1% of the highest scoring vertices as potential keywords. This gives us a number of keywords ranging from 8 up to 51 per document. Additionally, since up to this point the list of potential keywords only consists of single word entities, the authors advise that key-phrases can also be generated from this list. This is done by combining two unique keywords with one another. To validate such key-phrases, the mutated entities are checked against the entire content of the document. This determines whether such a phrase indeed exists in the text.

5. RESULTS

Thus, upon selecting, constructing and implementing the necessary artefacts, a collection of results have been retrieved from each executed technique. These results are a direct output of all statistical and linguistic computations that the textual content underwent. This makes the generated output available for evaluation. Given that the study deals with domain-specific documents, domain knowledge was needed to also evaluate the results. Fortunately, as mentioned earlier, competent entities within the bank were asked to evaluate the algorithmic outcomes. This was mostly true for automatic summarization and keyword extraction, whereas the reference extraction did not necessarily require domain specific understanding to be evaluated.

5.1. Reference Extraction Results

The relatively small corpus of documents enabled the creation of a golden standard that contained all the referencing policies for each policy document. This golden standard held the entire collection of true values, against which values the algorithmic results were evaluated. Thus, based on what the outcome was, we were able to compute the accuracy of the algorithm with the use of *precision (P), recall (R)* and *F-measure (F)* (Sasaki, 2007). *P* and *R* have been regularly used to measure the performance of information retrieval and information extraction systems (Makhoul & Kubala, 1999), by determining the success rate of the algorithm. As such, *P* (also known as *confidence*) determines how many of the retrieved values are indeed correctly predicted. Whereas *R*, also known as *sensitivity*, determines how many values from the gold standard are correctly predicted by the algorithm. Nevertheless, the scientific community decided to combine these performance indicators under a single measure of performance, thus introducing *F*, which is an equally weighting equation of both *P* and *R*. Additionally,

researchers such as (Powers, 2011) argue that in IE experiments related to Machine Learning and Computational Linguistics, more importance should be put on determining how confident one can be with the rules or classifier. This means that when measuring the accuracy of the algorithm, more weight should be put on *P*. In such cases, a variation of *F* is used, namely the F_β formula. This variation of the *F* formula usually takes two values as β. It takes a value of 0.5 when more weight should be put on *P*, and a value of 2 when more weight should be put on *R*. Hence, this study evaluates the results based on the normal weighted *F* formula and $F_{0.5}$ formula. The results from these calculations are given in Table 1, together with all the relevant details for each policy. According to these equations,

Table 1. Reference extraction results

Title	Total	Correct	Part.Corr.	Miss	Extra	P	R	F	Fβ
Policy A	15	11	3	1	0	1	0.73	0.84	0.93
Policy B	15	15	0	0	0	1	1	1	1
Policy C	13	8	4	1	0	1	0.61	0.76	0.88
Policy D	11	4	1	6	0	1	0.36	0.53	0.73
Policy E	13	12	1	0	2	0.85	0.92	0.88	0.86
Policy F	18	13	4	1	0	1	0.72	0.83	0.92
Policy G	14	12	2	0	1	0.92	0.85	0.88	0.90
Policy H	11	11	0	0	0	1	1	1	1
Policy I	27	21	5	1	0	1	0.77	0.87	0.94
Policy J	31	26	4	1	0	1	0.83	0.9	0.96
Policy K	6	6	0	0	0	1	1	1	1
Policy L	20	14	6	0	0	1	0.7	0.82	0.92
Policy M	13	11	2	0	0	1	0.84	0.91	0.96
Policy N	13	13	0	0	0	1	1	1	1
Policy O	24	22	2	0	0	1	0.91	0.95	0.98
Policy P	17	14	0	3	0	1	0.82	0.90	0.95
Policy Q	8	8	0	0	0	1	1	1	1
Policy R	10	10	0	0	0	1	1	1	1
Policy S	11	10	1	0	0	1	0.90	0.94	0.97
Policy T	77	74	1	2	1	0.98	0.96	0.97	0.97
Policy U	29	21	6	2	0	1	0.72	0.83	0.92
Policy V	22	16	2	4	0	1	0.72	0.84	0.92
Policy W	17	16	0	1	0	1	0.94	0.96	0.98
Total	435	368	44	23	4	22.75	19.3	20.61	21.63
Average	18.91	16	1.91	1	0.17	0.99	0.84	**0.89**	**0.94**

the algorithm managed to achieve a 89% accuracy in extracting relevant information, when both of the performance indicators were weighted equally, and a 94% accuracy when the performance was measured with the F_β formula. Both of these results showed that the algorithm was highly capable to extract relevant information. It should be mentioned that in some cases the algorithm extracted

imperfect chunks. These imperfect chunks, mostly, missed a part of the information, since it contained entities expressed in a language different than English. In order to determine how to deal with such imperfect chunks, the literature was consulted. This, showed that the concept of a mistake/negative (*i.e.* partially correct) is not properly defined and it is a subject of change depending on researcher's perception (Hripcsak & Rothschild, 2005). Thus in the initial evaluation, the imperfect information were considered as entirely incorrect (negative) values. Furthermore, a second evaluation took place, where these partially correct chunks were considered as correct. This evaluation managed to yield an accuracy of 95% and 97%, when calculating F and F_β respectively. This confirms that the constructed artefact performed impressively.

5.2. Summary Extraction Results

Given the domain specific dictionary that is used to compile these documents, the evaluation of the summaries required matching domain knowledge to assess them. Henceforth, domain experts were charged with the responsibility to evaluate these summaries. The experts were asked to assess the summary and to comment on issues such as whether the summary covered the main aspects of the document, whether the summary was a good representation of the policy, what did the summary miss, and other comments of the similar nature. As mentioned earlier, such a form of evaluation followed a somewhat Black-Box evaluation approach where a comment was provided regarding the accuracy of the outcome. Through an evaluation form the experts could read the generated summary for the respective policy, and provide a comment about its ability to convey the main message of the document. Although the algorithm was capable to construct summaries, the expert evaluation showed that not always the generated outcome was adequate. In some scenarios the generated output had some data quality issues, like having the tendency to generate a more detailed summary than necessary. Nevertheless, next to it, there were also summaries that were a good representation of the policy, as well as a moderate representation of the policy. These comments were equally dispersed across the evaluated corpus, enabling us to sum them up into three main thematic representations:

- The summary covers the mains aspect of the policy found in 7 evaluations (30.4%);
- The summary is a moderate representation of the policy found in 8 evaluations (34.7%);
- The summary is too detailed, thus does not cover the aspect of the policies found in 8 evaluations (34.7%).

5.3. Keyword Extraction Results

The extracted keywords also required some domain knowledge to be properly evaluated. Thus, together with the summaries, the experts also evaluated the keywords that the algorithm generated for each policy. The experts annotated the algorithmic keywords either as relevant or irrelevant. At the same time, they were advised to add potential keywords or key-phrases that the algorithm had failed to recognize as such. Although in essence, it followed the same evaluation logic as with the summaries, when it came to evaluating keywords, this method enabled to translate the qualitative evaluation into quantitative representations. Such a transformation enabled to measure the performance of the algorithm with the use of P, R and F. The results from such an evaluation are provided in Table 2, together with all the relevant details for each policy. Furthermore, it showed that the algorithm managed to reach an average accuracy of 83% in extracting relevant keywords or key-phrases.

Nevertheless, considering that our list of keywords ranged from 8 up to 51 potential keywords per policy, one may argue that this is beyond a reasonable amount. Thus, looking at similar studies, it was noticed that most of them extracted 9 up to 15 keywords per policy (Hulth & Megyeesi, 2006; Mihalcea & Tarau, 2004; Rose, Engel, Cramer, & Cowley, 2010; Yang, Chen, Cai, Huang, & Leung, 2016). Henceforth, from our lists of keywords, the ones that exceeded this range, were capped at a maximum of 15 most representative keywords. These changes called for a re-evaluation of the

Table 2. Keyword extraction results

Title	Total	Correct	Incorrect	Suggested	P	R	F
Policy A	8	4	4	2	0.50	0.67	0.57
Policy B	13	8	5	0	0.62	1	0.76
Policy C	20	18	2	3	0.90	0.86	0.88
Policy D	29	23	6	2	0.79	0.92	0.85
Policy E	16	12	4	7	0.75	0.63	0.69
Policy F	14	10	4	2	0.71	0.83	0.77
Policy G	10	8	2	1	0.80	0.89	0.84
Policy H	19	11	8	0	0.58	1	0.73
Policy I	29	24	5	3	0.83	0.89	0.86
Policy J	51	43	8	10	0.84	0.81	0.83
Policy K	14	10	4	6	0.71	0.63	0.67
Policy L	13	10	3	2	0.77	0.83	0.80
Policy M	12	11	1	0	0.92	1	0.96
Policy N	13	11	2	0	0.85	1	0.92
Policy O	35	29	6	0	0.83	1	0.91
Policy P	13	12	1	0	0.92	1	0.96
Policy Q	10	7	3	0	0.70	1	0.82
Policy R	11	9	2	6	0.82	0.60	0.69
Policy S	17	16	1	0	0.94	1	0.97
Policy T	48	40	8	0	0.83	1	0.89
Policy U	13	12	1	0	0.92	1	0.96
Policy V	19	16	3	0	0.84	1	0.91
Policy W	11	9	2	0	0.82	1	0.90
Average	19	15	3.6	1.91	0.79	0.87	**0.83**

algorithm, which showed that if only the 15 highest ranking entities are concerned, the algorithm can reach an extraction accuracy of 82%, indicating that there is not much of a difference between the two calculations. To get a better understanding on how the algorithm performed in these two cases, we benchmarked the generated outcome with other similar studies (Hulth & Megyeesi, 2006; F Liu, Pennell, & Liu, 2009; Mihalcea & Tarau, 2004; Rose et al., 2010; Yang et al., 2016; Zhang et al., 2008). This benchmarking is shown in Table 3 and it indicates that the outcome of this experiment represents one of the highest accuracy levels that have been achieved when extracting keywords.

6. CASE STUDY IMPLICATIONS

What started as an attempt to determine the use of TM on business policies, more specifically bank policies, evolved into a novel approach of processing such text-heavy, domain-specific documents. Even though the available publications on policy documents gave little to no indication about which techniques should be used for the case study requirements, the rich body of literature in other domains pinpointed techniques that would attain the study objectives. This literature showed that when

Table 3. A comparative overview of keyword extraction studies

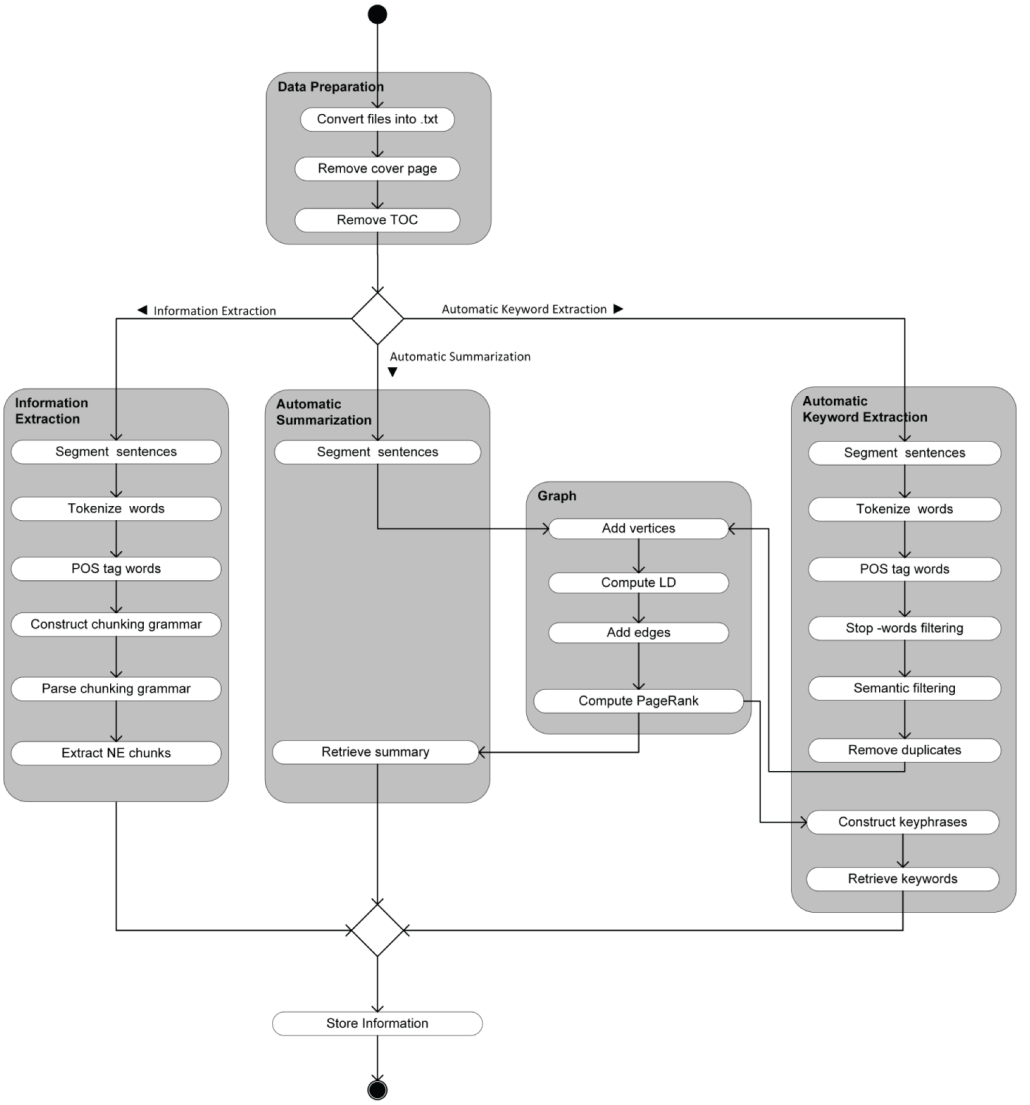
Study	P	R	F
Automatic Keyword extraction from individual documents (RAKE) (Rose et al., 2010)	33.7%	51.5%	37.2%
TextRank: Bringing order into text (Mihalcea & Tarau, 2004)	31.2%	43.1%	36.2%
Improved automatic keyword extraction given more linguistic knowledge (Yang et al., 2016)	22.5%	51.7%	33.9%
Unsupervised approach for automatic keyword extraction using meeting transcripts (F Liu et al., 2009)	NA	NA	19.6%
A study on automatically extracted keywords in Text Categorization (Hulth & Megyeesi, 2006)	92.89%	72.94%	81.72%
Automatic keyword extraction from documents using conditional random fields (Zhang et al., 2008)	66.3%	41.9%	51.25%
This Study	79.1%	89.3%	**83.2%**
This Study (max. 15)	79.5%	87.3%	82.3%

dealing with unstructured data, the processing can rely on a single or ensemble of TM techniques, depending on the intended outcome. Such was the nature of the case study at hand, where the set of requirements called for the use of multiple TM techniques. Looking thoroughly at the adaption of these techniques into programmable artefacts and the execution of the artefact on the study corpus, the algorithmic results did not fail to impress. The ability of the algorithm to reach an over 89% accuracy in extracting relevant information, hints strongly towards the potential of such techniques to process policy documents. A distinctive feature of this artefact was its ability to miss a relatively small portion of relevant information from the corpus. On a similar note, the automatic keyword extraction technique proved to be highly capable of recognizing and extracting the most distinctive words of a document, thus outperforming similar studies.

Although the summarization of the documents did not share the same level of success, it still managed to reach a moderate success. Furthermore, its tendency to generate more detailed summaries than necessary shows that the issues in this approach are mostly data quality related rather than methodological. The results of this study are the first of its kind when it comes to internal business policies. While most of the literature studies have yet to execute their conceptual frameworks and ratify their claims, the results of this study and their statistical evaluation validate the use of such an approach on business policies and make it distinguishable from other approaches. Considering the novelty of this approach, when it comes to bank policies, and at the same time its validation by experts and statistical methods, a new framework for processing internal business policies is derived. Figure 2 provides a visual representation of this framework. Here, all the utilized techniques are combined into a single representation which provides a step-by-step description of the followed approach. It guides the user from the data preparation phase towards actions that concern reference extraction, keyword extraction and automatic summarization. At the same time, such phases can be recognized in the framework as the main activities, each having a collection of sub-activities. In addition to these, the *Graph* activity is a shared entity among keyword extraction and automatic summarization.

Furthermore, the design of this framework draws upon the meta-algorithmic representation of the approach followed in the case study. (Marco Spruit & Jagesar, 2016) define meta-algorithmic modelling (MAM) as an "[...] engineering discipline where sequences of algorithm selection and configuration activities are specified deterministically for performing analytical tasks based

Figure 2. Meta-algorithmic model for processing business policies



on problem-specific data input characteristics and process preferences". This approach relies on the activity recipes of the case study, which are modelled with the use of method–engineering notations. Method-engineering is defined as "the engineering discipline to design construct and adapt methods, techniques and tools for the development of information systems" (Brinkkemper, 1996). By following this approach, the framework is depicted as a Process Deliverable Diagram (PDD), which uses a UML activity diagram to represent the processes and a UML class diagram to represent the deliverables (van de Weerd & Brinkkemper, 2008). In general, an activity diagram falls in the behavioral class of UML diagrams, thus being a representation that depicts the behavior of a module. Whereas a class diagram belongs to the structural class of UML diagrams, thus representing the structure of the entities in the module. Nevertheless, this framework only represents the processes, thus composed of only a UML activity diagram.

7. CONCLUSION

Motivated by the fact that Text Mining (TM) is being used extensively across industries, this study has focused on determining the use of such a discipline on business policies, and the benefits that it brings for such documents. This investigation was initiated with a review of the current state of literature that has treated such concepts. Nevertheless, the literature review showed that the use of TM to process policies was far from what it was anticipated to be, which answers RQ1. In a time when the use of TM on disciplines such as biomedicine, national security, finance and law is flourishing, the use of TM on internal business policies, and policies in general, still falls short on both qualitative and quantitative aspects. The amount of publications that treat such concepts was quite limited and failed to match the prominence of other domains. In addition to this, the available publications only considered privacy policies in their study, also known as "Terms and Conditions". This also revealed that the implementation of such techniques on business policies, had only scarcely been addressed yet by other researchers. Furthermore, this small corpus of publications introduced numerous conceptual frameworks, nevertheless, however the majority of them lacked a full scientific validation. This is also reflected in the existence of systems that use TM techniques to process policy documents. Even though most of the literature revolved around the idea of creating such systems, only a small portion embodied the existing frameworks into working modules. Furthermore, as far as the qualitative aspect is concerned, the processing of policy documents often failed to go beyond text categorization/classification, Information Extraction, and Topic Modelling techniques, which answers RQ2. Thus, besides providing an overview of the current state of literature, that has investigated the use of TM on business policies, this study also indicates the gaps in the literature where it contributes. The biggest contribution comes from the introduced Meta-Algorithmic Modelling (MAM) framework that has been implemented and fully validated on bank policies. Besides utilizing a set of unprecedented techniques on policy documents, it also gives a step-by-step receipe of how our Top-3 TM techniques (information extraction, automatic summarization and automatic keyword extraction) can be implemented and what benefits they yield, which answers RQ3.

8. DISCUSSION

The case study, together with the results that derived from its implementation, provide a glimpse into the possibilities that TM offers in analyzing and simplifying text-heavy documents such as policies. The better part of the framework showed to be highly capable in analyzing such unstructured data, yielding results that are quite impressive even when compared to similar studies. Nevertheless, trying to think about the bigger picture, it is a matter of discussion whether such an approach can be used in other circumstances. Regarding this issue, one should consider that although the framework was evaluated on domain-specific documents, its composition relies entirely on generic NLP modules. The fact that this framework only relies on such artefacts, and still manages to yield promising results, is an indication of what the answer might be. A potential generalizability also comes from the fact that textual content from the legal domain shares some similarities with policy text. This makes it reasonable to expect similar success if the given approach is adapted to the judicial domain. Nevertheless, in a time when the benefits of this framework for policy documents are apparent, the question is how significant such a framework can be for other domains. Looking at the literature on legal text, the summarization of such lengthy documents was a common practice. This, together with the use of IE for identifying relevant entities, were commonly used to acquire a timelier understanding of such documents. Additionally, the continuous increase in biomedical publications makes it quite difficult to be up to date with the latest developments in the field. Thus, the automatic summarization of these documents, together with the ability to extract relevant entities from the text (i.e. enzyme reaction, drug names, chemical reactions etc.) is seen as a smart solution that enables to follow the domain developments in a contemporary manner.

8.1. Limitations

As far as limitations are concerned, there are some areas that introduce some form of limitation to the study. Most of the limitations came as a result of the dataset upon which the experiments were conducted. Taking into consideration that organizations, especially the ones in the financial domain, operate under a large number of internal policies, one may argue that a dataset of 23 internal policies is relatively small. This paves the way for discussing whether similar results will be acquired in a larger dataset. Next to this, the result evaluation process also faced some limitations. Due to time constraints, the summaries could only be evaluated in a form of acceptance testing. Thus, one may argue that such an evaluation approach may yield biased results. A better approach would have been to rely on statistical methods for summary evaluation, nevertheless for the time being such an approach was not feasible. Furthermore, given the fact that this research was designated only for internal documents, it was limited to a single case study. This means that the derived results are applicable only for the organisation where the experiment took place, thus impacting the external validity of the framework. Nevertheless, as it was mentioned earlier, the artefact gives enough freedom to tune it, depending on the applicability circumstances. Furthermore, based on the expert evaluation, limitations were also found in the implementation of the artefacts. While key-phrases were only constructed from two words, a considerable amount of expert suggestions exceeded this word limit, indicating that the creation of longer key-phrases should be considered.

8.2. Future Work

Given that the proposed framework is still in its early days, we initially intended to test it on a larger dataset, thus also overcoming one of the limitations. It can be argued that as the dataset increases, the representation of relevant information tends to differ more and more. Thus, by testing it on a larger dataset, we want to see the impact of the dataset on the results. Next to this, we call for the framework to be tested in different settings. What we mean by this is that the framework should be tested in both a different industry and a different bank. This will not only impact its validity, but it will also show whether the framework is still capable of deriving promising results by relying solely on generic modules. Furthermore, in this study, the IE aspect was only concerned with the references between policies. Considering that internal policies hold an extensive amount of valuable information, we consider testing this part of the framework on different types of relevant information as well. At the same time, given that the aspect using TM on internal policy documents is not that cultivated, we suggest that the future work in this domain should also consider techniques other than the ones utilized in the framework. This will not only contribute to the field, but it will also show whether other TM techniques can be implemented in alignment with the introduced MAM framework. Finally, we restate our motivation for this research being to determine a way that will enable an easier navigation through policy documents. We have yet to create an interface that will enable the users to search for policies, using the set of information that were extracted in the study, and evaluate its efficiency.

REFERENCES

Ammar, W., Wilson, S., Sadeh, N., & Smith, N. A. (2012). *Automatic categorization of privacy policies: A pilot study.* School of Computer Science, Language Technology Institute, Technical Report CMU-LTI-12-019.

Anton, A., & Earp, J. (2004). A requirements taxonomy for reducing Web site privacy vulnerabilities. *Requirements Engineering*, *9*, 169–185. doi:10.1007/s00766-003-0183-z

Anton, A. I., & Earp, J. (2003). *The Lack of Clarity in Financial Privacy Policies and the Need for Standardization*. Retrieved from http://www.truststc.org/wise/articles2009/article4.pdf

Bholat, D., Hansen, S., Santos, P., & Schonhardt-Bailey, C. (2015). Text Mining for Central Banks. *Centre for Central Banking Studies Handbook*, *33*, 1–19.

Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. O'Reilly Media, Inc.

Blei, D. M. (2012). Introduction to Probabilistic Topic Modeling. *Communications of the ACM*, *55*(4), 77–84. doi:10.1145/2133806.2133826

Brinkkemper, S. (1996). Method engineering: Engineering of information systems development methods and tools. *Information and Software Technology, 38*(4), 275–280. 10.1016/0950-5849(95)01059-9

Brodie, C. A., Karat, C.-M., & Karat, J. (2006). An empirical study of natural language parsing of privacy policy rules using the SPARCLE policy workbench. *Proceedings of the Second Symposium on Usable Privacy and Security - SOUPS '06*, 8. doi:10.1145/1143120.1143123

Copeland, L. (2003). *A Practitioner's Guide to Software Test Design*. Artech House.

Costante, E., den Hartog, J., & Petkovic, M. (2012). What Websites Know About You. In DPM/SETOP (pp. 146–159). Academic Press.

Costante, E., Sun, Y., Petković, M., & den Hartog, J. (2013). A machine learning solution to assess privacy policy completeness:(short paper). In *Proceedings of the 2012 ACM workshop on Privacy in the electronic society* (pp. 91–96). Academic Press.

Count, F. O. G., Reading, F., & Personnel, E. (1975). *Derivation of new readability formulas (automated readability inde, fog count and flesch reading ease formula) for navy enlisted personel*. Academic Press.

Fan, W., Wallace, L., Rich, S., & Zhang, Z. (2006). Tapping the power of text mining. *Communications of the ACM*, *49*(9), 76–82. doi:10.1145/1151030.1151032

Flesch, R. (1948). A New Readability Yardstick. *The Journal of Applied Psychology*, *32*(3), 221–233. doi:10.1037/h0057532 PMID:18867058

Friedman, C., Johnson, S. B., Forman, B., & Starren, J. (1995). Architectural requirements for a multipurpose natural language processor in the clinical environment. *Proceedings of the Symposium on Computer Applications in Medical Care*, 347–351.

Grimes, S. (2008). *Unstructured data and the 80 percent rule*. Carabridge Bridgepoints.

Grover, C., Matheson, C., Mikheev, A., & Moens, M. (2000). LT TTT - A Flexible Tokenisation Tool. *Proc. LREC 2000*. Retrieved from http://www.ltg.ed.ac.uk/papers/

Gunning, R. (1952). *The technique of clear writing*. Academic Press.

Haug, P. J., Ranum, D. L., & Frederick, P. R. (1990). Computerized extraction of coded findings from free-text radiologic reports. Work in progress. *Radiology*, *174*(2), 543–548. doi:10.1148/radiology.174.2.2404321 PMID:2404321

Heeringa, W., Nerbonne, J., Van Bezooijen, R., & Spruit, M. R. (2007). Geography and population size as explanatory factors for variation in the Dutch dialectal area. *Tijdschrift Voor Nederlandse Taal-En Letterkunde, 123*(1).

Hevner, A. R., March, S. T., Park, J., Ram, S., & Ram, S. (2004). Research Essay Design Science in Information. *Management Information Systems Quarterly*, *28*(1), 75–105. doi:10.2307/25148625

Hripcsak, G., & Rothschild, A. S. (2005). Agreement, the F-measure, and reliability in information retrieval. *Journal of the American Medical Informatics Association*, *12*(3), 296–298. doi:10.1197/jamia.M1733 PMID:15684123

Hulth, A., & Megyeesi, B. B. (2006). A study on automatically extracted keywords in text categorization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics* (Vol. 1, pp. 353–360). doi:10.3115/1220175.1220243

Kanapala, A., Pal, S., & Pamula, R. (2019). Text summarization from legal documents: A survey. *Artificial Intelligence Review*, *51*(3), 371–402. doi:10.1007/s10462-017-9566-2

Larson, M. (2012). *Automatic Summarization. Foundations and Trends® in Information Retrieval* (Vol. 5). 10.1561/1500000020

Levenshtein, V. I. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics, Doklady*, *10*(8), 707–710.

Li, J., Wang, H. J., Zhang, Z., & Zhao, J. L. (2010). A policy-based process mining framework: Mining business policy texts for discovering process models. *Information Systems and e-Business Management*, *8*(2), 169–188. doi:10.1007/s10257-009-0112-x

Liu, F., Ramanath, R., Sadeh, N., & Smith, N. (2014a). *A Step Towards Usable Privacy Policy: Unsupervised Alignment of Privacy Statements*. Retrieved from http://www.coling-2014.org/accepted-papers/585.php

Liu, F., Ramanath, R., Sadeh, N., & Smith, N. A. (2014b). A step towards usable privacy policy: Automatic alignment of privacy statements. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers* (pp. 884–894). Academic Press.

Liu, F., Pennell, D., & Liu, Y. (2009). Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics* (pp. 620–628). doi:10.3115/1620754.1620845

Makhoul, J., & Kubala, F. (1999). *Performance measures for information extraction*. Retrieved from https://books.google.com/books ?hl=en&lr=&id= uuR3mpBI5ksC&oi= fnd&pg=PA249&dq= Performance+measures+for+information+extraction&ot s=DN2Am7TIcT&sig= 47QBj2yOUss ENGUPKO nbJETOPBI

Marcus, M. P., Santorini, B., & Marcinkiewicz, M. A. (1993). Building a large annotated corpus of English: The Penn Treebank. *Computational Linguistics*, *19*(2), 313–330. doi:10.1162/coli.2010.36.1.36100

Massey, A. K., Eisenstein, J., Antón, A. I., & Swire, P. P. (2013). *Automated Text Mining for Requirements Analysis of Policy Documents*. Retrieved from https://www.cc.gatech.edu/~jeisenst/papers/re13rt-p085-p-18125-preprint.pdf

McLaughlin, G. H. (1969). SMOG grading: A new readability formula. *Journal of Reading*, *12*(8), 639–646. doi:10.1039/b105878a

Menger, V., Scheepers, F., & Spruit, M. (2018). Comparing Deep Learning and Classical Machine Learning Approaches for Predicting Inpatient Violence Incidents from Clinical Text. *Applied Sciences (Basel, Switzerland)*, *8*(6), 981. doi:10.3390/app8060981

Michael, J. B., Ong, V. L., & Rowe, N. C. (2001). Natural-language processing support for developing policy-governed software systems. *Proceedings 39th International Conference and Exhibition on Technology of Object-Oriented Languages and Systems. TOOLS 39*, 263–274. doi:10.1109/TOOLS.2001.941679

Mihalcea, R., & Tarau, P. (2004). *TextRank: Bringing order into texts*. Association for Computational Linguistics.

Miner, G., Elder, I. V. J., Fast, A., Hill, T., Nisbet, R., & Delen, D. (2012). *Practical text mining and statistical analysis for non-structured text data applications*. Academic Press.

Niu, R., & Shen, B. (2019). Microblog User Interest Mining Based on Improved TextRank Model. *Journal of Computers*, *30*(1), 42–51.

Page, L., & Brin, S. (1997). *PageRank: Bringing order to the web.* Stanford Digital Libraries Working Paper, 72.

Powers, D. M. W. (2011). Evaluation: From Precision, Recall and F-Measure To Roc, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies*, *2*(1), 37–63.

Rajaraman, A., & Ullman, J. D. (2011). Data Mining. *Mining of Massive Datasets*, *18*(Suppl), 114–142. doi:10.1007/978-1-4419-1280-0

Ramanath, R., Liu, F., Sadeh, N., & Smith, N. A. (2014). *Unsupervised alignment of privacy policies using hidden Markov models*. Retrieved from http://repository.cmu.edu/lti/150/

Renz, I., Ficzay, A., & Hitzler, H. (2003). Keyword Extraction for Text Characterization. *8th International Conference on Applications of Natural Language to Information Systems*, 228–234.

Rose, S., Engel, D., Cramer, N., & Cowley, W. (2010). Automatic keyword extraction from individual documents. *Text Mining: Applications and Theory*. 10.1002/9780470689646.ch1

Sadeh, N., Acquisti, A., Breaux, T. D., Cranor, L. F., Mcdonald, A. M., Reidenberg, J. R., ... (2013). *The Usable Privacy Policy Project: Combining Crowdsourcing, Machine Learning and Natural Language Processing to Semi-Automatically Answer Those Privacy Questions Users Care About.* Tech. Report CMU-ISR-13-119, (1). Retrieved from http://ra.adm.cs.cmu.edu/anon/usr0/ftp/home/anon/isr2013/CMU-ISR-13-119.pdf

Sanner, M. F. et al. (1999). Python: A programming language for software integration and development. *Journal of Molecular Graphics & Modelling*, *17*(1), 57–61. PMID:10660911

Sasaki, Y. (2007). The truth of the F-measure. *Teach Tutor Mater*, 1–5. Retrieved from https://www.cs.odu.edu/~mukka/cs795sum09dm/Lecturenotes/Day3/F-measure-YS-26Oct07.pdf

Sibley, E. H., Michael, J. B., & Wexelblat, R. L. (1991). Use of an experimental policy workbench: Description and preliminary results. *Results of the IFIP WG 11.3 Workshop on Database Security V: Status and Prospects*, 47–76.

Soukoreff, R. W., & MacKenzie, I. S. (2001). *Measuring errors in text entry tasks*. 10.1145/634067.634256

Spence, D. (2010). Data, data everywhere: a special report on managing information. *The Economist*, 1–10. 10.1136/bmj.f725

Spruit, M., & Jagesar, R. (2016). Power to the People! - Meta-Algorithmic Modelling in Applied Data Science. *Proceedings of the 8th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management, 1*(Ic3k), 400–406. doi:10.5220/0006081604000406

Spruit, M., & Ferati, D. (2019). Applied Data Science in Financial Industry: Natural Language Processing Techniques for Bank Policies. In A. Visvizi & M. Lytras (Eds.), *Springer Proceedings in Complexity, Research & Innovation Forum 2019* (pp. 351–367). Springer.

Spruit, M., & Lytras, M. (2018). Applied data science in patient-centric healthcare: Adaptive analytic systems for empowering physicians and patients. *Telematics and Informatics*, *35*(4), 643–653. doi:10.1016/j.tele.2018.04.002

Spruit, M. R. (2006). Measuring syntactic variation in Dutch dialects. *Literary and Linguistic Computing*, *21*(4), 493–506. doi:10.1093/llc/fql043

Stamey, J. W., & Rossi, R. a. (2009). Automatically identifying relations in privacy policies. *Proceedings of the 27th ACM International Conference on Design of Communication - SIGDOC '09*, 233. doi:10.1145/1621995.1622041

Syed, S., Borit, M., & Spruit, M. (2018). Narrow lenses for capturing the complexity of fisheries: A topic analysis of fisheries science from 1990 to 2016. *Fish and Fisheries*, *19*(4), 643–661. doi:10.1111/faf.12280

van de Weerd, I., & Brinkkemper, S. (2008). Meta-modeling for situational analysis and design methods. Handbook of Research on Modern Systems Analysis and Design Technologies and Applications, 35.

Voutilainen, A. (2003). Part-of-speech tagging. In The Oxford handbook of computational linguistics. OUP.

Wilson, R. C., & Hancock, E. R. (2004). Levenshtein distance for graph spectral features. *Proceedings - International Conference on Pattern Recognition, 2*(C), 489–492. doi:10.1109/ICPR.2004.1334272

Witten, I. H. (2004). Text Mining. *International Journal of Computational Biology and Drug Design*, *198*.

Wohlin, C. (2014). Guidelines for Snowballing in Systematic Literature Studies and a Replication in Software Engineering. *18th International Conference on Evaluation and Assessment in Software Engineering (EASE 2014)*, 1–10. doi:10.1145/2601248.2601268

Xiao, X., Paradkar, A., & Xie, T. (2011). Automated extraction and validation of security policies from natural-language documents. *Perspective*, *11*. doi:10.1145/2393596.2393608

Xiong, C., Li, X., Li, Y., & Liu, G. (2018). Multi-documents summarization based on TextRank and its application in online argumentation platform. *International Journal of Data Warehousing and Mining*, *14*(3), 69–89. doi:10.4018/IJDWM.2018070104

Yang, K., Chen, Z., Cai, Y., Huang, D. P., & Leung, H. (2016). *Improved automatic keyword extraction given more semantic knowledge*. 10.1007/978-3-319-32055-7_10

Zhang, C., Wang, H., Liu, Y., Wu, D., Liao, Y., & Wang, B. (2008). Automatic Keyword Extraction from Documents Using Conditional Random Fields. *Journal of Computational Information*, *43*, 1169–1180. Retrieved from http://www.jofci.org

Zhao, Y. (2013). Analysing twitter data with text mining and social network analysis. *Proceedings of the 11th Australasian Data Mining and Analytics Conference (AusDM 2013)*.

Zhong, S., & Ghosh, J. (2005). Generative model-based document clustering: A comparative study. *Knowledge and Information Systems*, *8*(3), 374–384. doi:10.1007/s10115-004-0194-1

Marco Spruit (PhD) is an Associate Professor in the Natural Language Processing research group of the Department of Information and Computing Sciences within the Faculty of Science at Utrecht University. As principle investigator in the department's Applied Data Science Lab, his research investigates Natural Language Processing Systems for Self-Service Data Science. Marco's research objective for the coming years is to establish and lead an authoritative national infrastructure for Dutch natural language processing (NLP) to facilitate and popularise self-service data science.

Drilon Ferati has been working as an IT Development Engineer and Business Analyst at ABN AMRO Bank in The Netherlands.