A Cognitive Computing Approach for Classification of Complaints in the Insurance Industry

J Forster\textsuperscript{1, 2,*} and B Entrup\textsuperscript{2}

\textsuperscript{1}Department for Computer Sciences and Mathematics, University of Applied Sciences Munich, Lothstraße 64, 80335 Munich, Germany
\textsuperscript{2}Global Business Services, IBM Germany, IBM-Allee 1, 71139 Ehningen, Germany

* jforster@hm.edu

Abstract. In this paper we present and evaluate a cognitive computing approach for classification of dissatisfaction and four complaint specific complaint classes in correspondence documents between insurance clients and an insurance company. A cognitive computing approach includes the combination classical natural language processing methods, machine learning algorithms and the evaluation of hypothesis. The approach combines a MaxEnt machine learning algorithm with language modelling, tf-idf and sentiment analytics to create a multi-label text classification model. The result is trained and tested with a set of 2500 original insurance communication documents written in German, which have been manually annotated by the partnering insurance company. With a F1-Score of 0.9, a reliable text classification component has been implemented and evaluated. A final outlook towards a cognitive computing insurance assistant is given in the end.

1. Introduction
Every day we create 2.5 quintillion bytes of data in all kinds of industries and according to the widely quoted 80 % rule, most of this enterprise data is generated and stored in an unstructured form such as text and images [1]. This applies in particular to the insurance industry, where medical assessments, claim reports, bills, blanks and letters arrive every second. The process automation quota is growing steadily, for example for property/casualty insurers from 11.8 % in 2013 to 15.3 % in 2014 [2]. However, this automation relies on structured or at least semi-structured documents and data. Yet, companies are aware of the importance of their internal and external unstructured data, while only 25 % of unstructured data is used according to Forrester [1] as one can see in figure 1.
2. Problem Definition
A large part of the growing challenge regarding unstructured data in the insurance industry is related to the change of communication. With the ease of writing an email instead of a letter as well as the rise of mostly internet-dependent direct insurance companies comes an increase of digital correspondence. These have to be handled by the input management systems of these companies. While semi-structured data, such as bills and forms can be handled by rule-based classification systems, cover letters are completely unstructured. However, valuable information can be extracted from it and be used for a more efficient routing in an insurance company.

In this paper, a machine learning approach for handling unstructured insurance cover letters is evaluated on the example of complaint letters. As one can see in figure 2, the classification of an complaint has two stages. In the first stage, the letters without dissatisfaction have to be separated from the letters, which could contain a complaint. In the second stage, the reason of the complaint has to be identified to ensure better case handling. In the case of assigning a dissatisfaction type, we have
4 classes: 'contract related', 'time related', 'quality related', and 'monetary related'. There are two general strategies for solving such a multi-class classification problem: one-vs-all (OVA) and one-vs-one (OVO). While OVA requires one binary classifier classifying each class against all other classes (4 in our case), OVO requires one classifier classifying each combination of classes \(4\*(4 - 1) / 2 = 6\) in our case). We select OVA as our strategy, due to its conceptual simplicity and because it has been shown to be as accurate as any other approach if finely-tuned regularized classifiers are used [3].

2.1. Limitations of Current Solution
Currently deployed input management systems rely heavily on static rules. These can identify standardized formulas or even react to specific keywords in continuous text. Multiple problems arise when dealing with cover letters from customers:

- **Terminology:** Insurance customers in general have highly varying backgrounds and therefore use different words and syntax, especially within complaints. In the test set used in this paper and described in more detail later, lawyers for example complain in a more complex and extensive way about an incompliant process than a regular insurance holder who is unhappy with the quality of service. Furthermore, brokers who complain on behalf of their customers tend to use more professional insurance terms than regular customers.

- **Context:** Important or subliminal information is often hidden in the greater context of a cover letter. Especially dissatisfaction can be expressed in a subliminal form. The request for sending a contract cancelation confirmation can be written as “Please send me a written confirmation of the contract termination” or “This is now the 3rd time I have to remind you to send me the confirmation of the contract termination”. Both have the same intention but with a different sentiment.

- **Equivalence:** In domains where there is a special terminology used by professionals but also laypersons, words are meant differently as used by the different groups. For example, in the insurance context of this paper an insurance policy, insurance contract and an insurance note have different meanings for the professional clerks, but are treated as synonyms by the clients.

Still, all these ambivalent and unstructured messages have to be processed by input management systems, which are not able to satisfactorily distinguish between two similar sentences as shown above. Until now only human brains have the cognitive capability to handle unstructured data.

2.2. Cognitive Computing
This paper evaluates a solution to overcome the above mentioned problems with unstructured text. Kelly et al. introduced the concept of cognitive computing [4]. The philosophy behind cognitive computing is the combination of various technologies to ‘understand’ unstructured data such as text, images, sound and video like a human brain is capable of. Real cognitive computing consists of three columns as depicted in figure 3. First of all the unstructured data has to be made analysable for machines. Image recognition technologies are used while natural language processing makes text handling possible. The next column represents the human capability of learning with various machine learning techniques such as neuronal networks, deep learning algorithms or decision trees. The third and final column is the most challenging to achieve: the human capability of creating hypothesis and reason. Diverse statistical algorithms are designed, trained and tested for each single use case.
3. Related Work
The work presented in this article is related to the research areas of machine learning, natural language processing and document classification. These areas and the related work are used together to create a cognitive computing approach for document classification in a real life insurance environment and reduce the problems described in chapter two occurring in this environment.

3.1. Document Classification
The automatic classification of text has a long history in information technology research [5]. Text classification tasks can be divided into single-label (one category is assigned to each text) and multi-label (several categories can be assigned) cases. While there are also dynamic and static rule based classification systems, we want to concentrate on adaptive machine learning classification systems. Kailer et al. describe and evaluate different state of the art classification models but also highlight the ambivalence depending on each practical use case and the underlying data set [6]. The in this research used data set of insurance related correspondence has not yet been assessed. The creation of a new cognitive computing based text classification approach is the general purpose of this research.

3.2. Machine Learning
In the last decade, binary machine learning algorithms such as decision trees [7], probabilistic classifiers such as Naive Bayes [8], support vector machines [9], and maximum entropy (MaxEnt) classifiers have been applied successfully to the task of document classification [10]. In the case of multiclass classification, these are combined either according to a one-vs-one or one-vs-all scheme. Rifkin et al. show that the latter given finely-tuned binary classifiers is at least as accurate as more complex approaches. Recently, Kim has achieved state-of-the-art results on several text classification benchmark tasks using convolutional neural networks [11]. The integration of machine learning reduces the 'context' problem.

3.3. Sentiment Analysis
The field of Sentiment Analysis deals with extracting opinion, mood and attitude from unstructured data. This task can be performed on different granularity levels. When whole documents are the unit of analysis, this is carried out as classification where the labels are so-called polarities e.g. positive or negative [12].

A popular approach is the use of a lexicon containing words annotated with these polarities. A more challenging problem arises when one tries to grade the strength of the opinion on a numeric scale (rating inference problem) [13].

We use simple heuristics to tackle this problem. They make use of the lexicon to infer the rating for a sentence from its syntax. These ratings are then combined to form a document rating. The usage of sentiment analysis also reduces the 'context' problem.
3.4. **Language Modelling**
Latent dirichlet allocation (LDA) or topic modelling is a generative process that allows the discovery of latent themes that underlie a set of documents [14]. Topic modelling algorithms have been adapted and used in a wide array of applications, such as to find patterns in genetic data, images, and social networks, but are particularly popular in information retrieval, classification, and corpus exploration tasks as they allow the organization of a large collection of unstructured texts according to its main topics. Topic models have been extended, e.g. to consider the change of topics over time [14], hierarchies of topics [15], as well as metadata such as authors [16] or inter-document links [17]. Language Modelling can lessen the 'terminology' issue.

3.5. **Tf-idf**
Term frequency-inverse document frequency (tf-idf) is a numerical method to compute the relevance of a term in a document given a corpus of documents. For a word we use its raw frequency ($tf_t$) as well as the so called "inverse document frequency" ($idf_t$), which was introduced 1972. The latter measures the significance of a term $t$ in a corpus by comparing the document frequency $df_t$, which is the number of documents that contain $t$, and the total number of documents $N$. Combined, these notions yield a composite value for each term in a document:

\[
tf - idf_t = tf_t \times idf_t = tf_t \times \log \frac{N}{df_t}
\]

See [18] for more information. Implementing tf-idf in our architecture tackles the 'equivalent' problem.

4. **Cognitive Computing Approach**
To evaluate the usefulness and feasibility of a Cognitive Computing document classifier we used the above mentioned technologies in combination to fulfil two of the three steps towards a cognitive computing system.

The implanted UIMA-based dissatisfaction classifier classifies between dissatisfaction and non-dissatisfaction as well as between different dissatisfaction types using a binary classifier for each. It can be trained and tested on two disjoint sets or can be evaluated using cross-validation on a single set of documents. It employs various lexical and semantic features as well as topic models and word features for classification.

4.1. **UIMA**
The UIMA (Unstructured Information Management Architecture) standard has been developed by IBM and is the defacto standard for analysing unstructured data. The basic architecture is built as an UIMA pipeline as depicted in figure 4.

![Figure 4. Classical UIMA pipeline](image)

The text elements or documents are handled from left to right, with basic natural language capabilities in the beginning to more sophisticated and use case specific modules at the end [19].
4.2. Orchestration
On top of the UIMA architecture we use the ClearTK toolkit, developed at the University of Colorado. It is a natural language processing framework which offers interfaces to different machine learning libraries [20].

5. Experiment
In the following we are training and evaluating a MaxEnt algorithm on documents manually annotated by domain experts. This documents consist of emails, letters and faxes which have been actually sent by insurance customer to the insurance company cooperating in our research project.

5.1. Dataset
For training and evaluation, domain experts manually annotated a training set of 2,500 documents and a test set of 1,500 documents (This almost evenly split set was a requirement of the involved insurance company). The training set contains 1,500 texts with complaints and 1,000 non-complaints. Both sets contain documents that have been assigned one or more labels indicating the type of complaint expressed in the text. Each text can contain up to four different classes since one text can express several complaints or one complaint can belong to several of the possible classes.

5.2. Learning and Testing Procedure
Before starting the actual feature extraction and training, a number of pre-processing steps were applied. First, the texts were cleaned: header and footers of e-mails and letters were removed, and the texts were anonymized. Furthermore, the texts were tokenized, lemmatized, and part-of-speech (POS) tagging was applied. Then the actual features were extracted. Besides a bag-of-words approach using all lemmas, synonyms were inserted, bi-grams extracted, and the lemmas with the highest tf-idf and log-likelihood values were calculated and sorted by their POS. Besides these features, each text was assigned a set of topic models and the sentiment model described above was used to calculate first a value for each word and combine these values for an overall value for each text. Some other features based on pattern matching, e.g., the use of caps-lock and the number of exclamation marks, were also used but will not be discussed and will not be evaluated in detail here. In total, five different classes were investigated: does the text contain a complaint? If so, does it concern the quality or the time of the treatment or processing, does it concern the contract conditions, or does it concern the payment? The latter four are core type of complaints and each text can be assigned more than one of these classes.

5.3. Evaluation
Although the project described here included several sub-classes that were successfully implemented, only the binary classification complaint/no-complaint classification can be considered and evaluated here. The overall final results that were obtained are very satisfying. Complaints could be identified with a precision of 0.84 and a recall of 0.97, resulting in an f1-score of 0.90. To evaluate the impact of the chosen features on the final results, an ablation study is performed, i.e., each feature is removed one after another, the model is trained and evaluated again, and the difference, i.e., the impact, of leaving out a single feature is measured by the difference in f1-score.

| Feature        | Precision | Recall | F1    | Differences |
|----------------|-----------|--------|-------|-------------|
| Baseline       | 0.84      | 0.97   | 0.90  | 0.00        |
| Bag-of-words   | 0.59      | 0.87   | 0.70  | -0.20       |
| Synonyms       | 0.84      | 0.97   | 0.90  | 0.00        |
As can be seen from table 1, most of the presented features contribute at least partially to the overall performance. The bag-of-words approach has the highest impact if left out. The remaining features only increase the f1-score by 0.2 points. Additional information is also added by the tf-idf based features, i.e., the highest scoring words of the POS noun, verb, and adjective. Moreover, a large impact can be found when leaving out the regex-based features. These patterns are based on experience and a looking for keywords that seem to be indicating a possible complaint. The topic models and the sentiment score based on a sentiment dictionary especially tuned to the insurance sector contribute to the final model but to a lower degree. The domain specific tuning has to be done because standardized sentiment analysis tend to give an insurance related letter a negative rating as they are often related to damage, losses, illness etc. However this is the daily business of the insurance industry and cannot be treated as a negative sentiment.

6. Conclusion
The promising first results depicted in table 1 lead now to an ongoing research activity within the insurance company as well with and integration into the current document process flow. Furthermore the cognitive computing approach will be used for enhancing the user experience at the workplace of each insurance clerk handling client cases. A first step has been made by implementing a text summary tool based on the text rank algorithm [21].This leads to a cognitive advisor tool for insurance clerks in the near future as the proven cognitive computing approach can now be adopted for more detailed domain specific document classes. Additionally the domain specific sentiment analysis will be evaluated and enhanced in a separate research approach.

References
[1] B. Evelson, L. Owens und B. Hopkins, Market Overview: Text Analytics, 2015.
[2] M. Hundt, Versicherer-Kommunikation: Papier dominiert weiter, 2015.
[3] R. Rifkin und A. Klautau, In defense of one-vs-all classification, The Journal of Machine Learning Research, Bd. 5, pp. 101-141, 2004.
[4] J. Kelly III und S. Hamm, Smart Machines: IBM's Watson and the Era of Cognitive Computing, Columbia University Press, 2013.
[5] H. Borko und M. Bernick, Automatic document classification, Journal of the ACM (JACM), Bd. 10, pp. 151-162, 1963.
[6] D. Kailer, P. Mandl und A. Schill, Grouping Product Aspects from Short Texts Using Multiple Classifiers, in Web Information Systems Engineering – WISE 2015, Springer International Publishing, 2015, pp. 1-15.
[7] W. W. Cohen und H. Hirsh, Joins that Generalize : Text Classification Using WHIRL, KDD-98 Proceedings, pp. 169-173, 1998.
[8] D. Koller und M. Sahami, Hierarchically Classifying Documents Using Very Few Words, Proceedings of the Fourteenth International Conference on Machine Learning, pp. 170-178, 1997.
[9] T Joachims, Text Categorization with Suport Vector Machines: Learning with Many Relevant Features, Proceedings of the 10th European Conference on Machine Learning ECML '98, pp. 137-142, 1998.
[10] H. L. Chieu und H. T. Ng, A maximum entropy approach to information extraction from semi-structured and free text, AAAI/IAAI, Bd. 2002, pp. 786-791, 2002.
[11] Y. Kim, „Convolutional Neural Networks for Sentence Classification,“ 2014.
[12] Liu, Sentiment Analysis Mining Opinions, Sentiments, and Emotions, Cambridge University Press, 2015.
[13] B. Pang und L. Lee, Seeing Stars: Exploiting Class Relationships for Sentiment Categorization with Respect to Rating Scales, in Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Stroudsburg, 2005.
[14] D. M. Blei, A. Y. Ng und M. I. Jordan, Latent Dirichlet Allocation, Journal of Machine Learning Research, Bd. 3, pp. 993-1022, 2003.
[15] D. Blei und J. Lafferty, Dynamic topic models, Proceedings of the 23rd international conference {ldots}, pp. 113-120, 2006.
[16] M. Rosen-Zvi, T. Griffiths, M. Steyvers und P. Smyth, The author-topic model for authors and documents, Proceedings of the 20th conference on Uncertainty in artificial intelligence, pp. 487-494, 2004.
[17] J. Chang und D. M. Blei, Hierarchical relational models for document networks, The Annals of Applied Statistics, Bd. 4, pp. 124-150, 2010.
[18] C. D. Manning, P. Raghavan und H. Schütze, Scoring, term weighting and the vector space model, Introduction to Information Retrieval, Bd. 100, pp. 2-4, 2008.
[19] D. Ferrucci und A. Lally, UIMA: an architectural approach to unstructured information processing in the corporate research environment, Natural Language Engineering, Bd. 10, pp. 327-348, 2004.
[20] P. V. Ogren, P. G. Wetzler und S. Bethard, ClearTK: A UIMA toolkit for statistical natural language processing, Towards Enhanced Interoperability for Large HLT Systems: UIMA for NLP, Bd. 32, 2008.
[21] R. Mihalcea und P. Tarau, TextRank: Bringing order into texts, 2004.