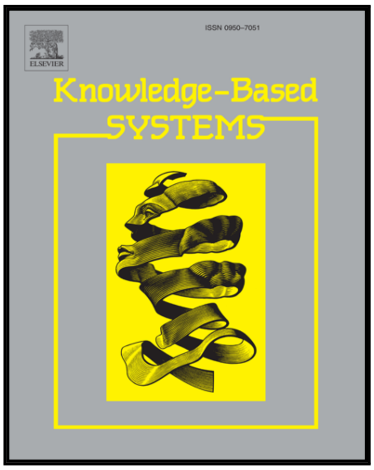
Accepted Manuscript



Text Normalization and Semantic Indexing to EnhanceInstant Messaging and SMS Spam Filtering

Tiago A. Almeida, Tiago P. Silva, Igor Santos,
José M. Gómez Hidalgo

PII: S0950-7051(16)30090-9
DOI: 10.1016/j.knosys.2016.05.001
Reference: KNOSYS 3498

To appear in: *Knowledge-Based Systems*

Received date: 30 October 2015
Revised date: 25 April 2016
Accepted date: 6 May 2016

Please cite this article as: Tiago A. Almeida, Tiago P. Silva, Igor Santos, José M. Gómez Hidalgo, Text Normalization and Semantic Indexing to EnhanceInstant Messaging and SMS Spam Filtering, *Knowledge-Based Systems* (2016), doi: 10.1016/j.knosys.2016.05.001

Text Normalization and Semantic Indexing to Enhance Instant Messaging and SMS Spam Filtering

Tiago A. Almeida[a,*], Tiago P. Silva[a], Igor Santos[b], José M. Gómez Hidalgo[c]

[a] *Department of Computer Science, Federal University of São Carlos (UFSCar), Sorocaba, São Paulo, 18052-780, Brazil*

[b] *Universidad de Deusto, DeustoTech – Computing (S3Lab), Avenida de las Universidades 24, Bilbao, Vizcaya, 48007, Spain*

[c] *Analytics Department, Pragsis, Manuel Tovar 49-53, Madrid, 28034, Spain*

Abstract

The rapid popularization of smartphones has contributed to the growth of online Instant Messaging and SMS usage as an alternative way of communication. The increasing number of users, along with the trust they inherently have in their devices, makes such messages a propitious environment for spammers. In fact, reports clearly indicate that volume of spam over Instant Messaging and SMS is dramatically increasing year by year. It represents a challenging problem for traditional filtering methods nowadays, since such messages are usually fairly short and normally rife with slangs, idioms, symbols and acronyms that make even tokenization a difficult task. In this scenario, this paper proposes and then evaluates a method to normalize and expand original short and messy text messages in order to acquire better attributes and enhance the classification performance. The proposed

*Corresponding author

Email addresses: talmeida@ufscar.br (Tiago A. Almeida), tpsilva@acm.org (Tiago P. Silva), isantos@deusto.es (Igor Santos), jmgomez@pragsis.com (José M. Gómez Hidalgo)

text processing approach is based on lexicographic and semantic dictionaries along with state-of-the-art techniques for semantic analysis and context detection. This technique is used to normalize terms and create new attributes in order to change and expand original text samples aiming to alleviate factors that can degrade the algorithms performance, such as redundancies and inconsistencies. We have evaluated our approach with a public, real and non-encoded dataset along with several established machine learning methods. Our experiments were diligently designed to ensure statistically sound results which indicate that the proposed text processing techniques can in fact enhance Instant Messaging and SMS spam filtering.

Keywords: Instant messaging spam filtering, SMS spam filtering, SPIM, text categorization, natural language processing

1. Introduction

Short text messaging is the mean of communication for a huge number of people nowadays. In this context, online Instant Messaging (IM) and SMS are clearly the leading technologies. In fact, it is estimated that about 80 billion messages are sent a day considering just SMS, WhatsApp and Facebook Messenger[1].

SMS has become a massive commercial industry since messaging still dominates mobile market non-voice revenues worldwide. According to a Portio Research report[2], the worldwide mobile messaging revenue was over 128 billion dollars in 2011, and in 2016 the revenue is forecasted to be over 153 billion dollars. The same document indicates that, in 2011, more than 7.8

[1]SMS, Messenger and WhatsApp process 80 billion messages a day. Available at `http://goo.gl/HdWm1v`.

[2]Mobile Messaging Futures 2012-2016. Available at `http://goo.gl/Wfb01z`.

trillion SMS messages were sent over the world, while more than 9.5 trillion were disseminated just in 2014.

In the same way, the popularization of smartphones along with low cost Internet plans are leading online Instant Messaging applications to become the means of electronic communication most used in the world. To get an idea, WhatsApp recently claimed to have over 1 billion users. Proportionally, this means that one in seven people on the planet use the messaging app[3]. According to a report released by Facebook in April 2016, about 70% of WhatsApp users access the application daily and more than 42 billion messages are sent a day. Moreover, Facebook Messenger has about 900 million monthly active users responsible for sending around 18 billion messages a day.

The growth in short text messaging along with unlimited texting plans allows malicious messages barely costs nothing for the attackers. This, combined with the trust users inherently have in their mobile devices, makes it a propitious environment for attack. As a consequence, Instant Messaging applications and SMS are becoming the latest target of electronic junk mail.

SMS spam (also known as mobile phone spam) is any junk message delivered to a mobile phone as text messaging. This practice, which became very popular in some parts of Asia, is now spreading in Western countries[4]. Besides being annoying, SMS spam can also be expensive since some users must pay to receive messages. Moreover, there is a very limited availability of mobile phone spam-filtering software and another concern is that important legitimate messages such as those of an emergency nature could be blocked. Nonetheless, many providers offer their subscribers means for mitigating unsolicited SMS messages.

More recently, the volume of spam is also increasing in similar environ-

[3]F8 – Facebook Developer Conference (April 2016), available at `http://goo.gl/HdWm1v`

[4]Cloudmark annual report. Available at `http://goo.gl/5TFAMM`.

ments. There are several indications that online Instant Messaging apps are the next target. Such messages are also known as SPIM – SPam over Instant Messaging. For instance, there is some evidence of chain letters and hoax in WhatsApp. Panda Labs have reported some of the most popular hoax in WhatsApp in Spain, in 2015, like the one that promises new emoticons if you click and send the same hoax (spam) to ten friends[5]. Acording to the company AdaptativeMobile, there are also spam campaigns in UK targeting WhatsApp users with investment spam messages sent from US numbers to Europe, spam promoting fake luxury goods sent from Chinese numbers to users in Europe, and others[6]. To avoid these messages, Facebook Messenger has added a feature to report a message as spam[7] and Skype has been reported by users as spammy as well[8].

In traditional e-mail spam problem, simple techniques as blacklisting are often used to complement the content-based spam filtering. These solutions block e-mails from certain senders, whereas whitelisting [31] delivers e-mail from specific senders to reduce the number of misclassified ham e-mails. DNS blacklisting is one particular solution that checks the host address against a list of networks or servers known to distribute spam [33, 49]. However, in IM and SMS spam domain, it is very difficult to having access to such data mainly because the providers must preserve the confidential data of their customers.

While companies are facing many problems in dealing with texting spam problem, academic researchers in this field are also experiencing difficulties. One of the concerns is that established email spam filters have their performance seriously degraded when used to filter SPIM or mobile phone spam.

[5]Las 5 estafas de WhatsApp más famosas de 2015. Available at `http://goo.gl/LY9gm7`.

[6]Spammers set their sights on WhatsApp that's that ruined then. Available at `http://goo.gl/yyya7D`.

[7]How do I report a message as spam? Available at `https://goo.gl/9qjkIr`.

[8]Spoofed message from contact. Available at `http://goo.gl/fw5wl4`.

This happens due to the small size of these messages. Furthermore, these messages are usually rife of slangs, symbols, emoticons and abbreviations that make even tokenization a difficult task.

Noise in text messages can appear in different ways. The following phrase is an example: *"Plz, call me bak asap... Ive gr8 news! :)"*. There are misspelled words *"Plz, bak, Ive, gr8"*, abbreviation *"asap"* and symbol *":)"*. In order to transcribe this phrase to a proper English grammar, a *Lingo* dictionary[9] would be needed along with a standard English dictionary, which associates each slang, symbol or abbreviation to a correct term. After a step of text normalization, the input phrase would be transcribed to *"Please, call me back as soon as possible... I have great news! :)"*.

In addition to noisy messages, there are other well-known problems such as ambiguous words in context (polysemy) and different words with the same meanings (synonymy), that can harm the performance of traditional machine learning techniques when applied to text categorization problems.

Both synonymy and polysemy can have their effect minimized by semantic indexing for word sense disambiguation [45, 54]. Such approaches associate meanings extracted from dictionaries to words by finding similar terms given the context of a message. In general, the effectiveness of using such dictionaries relies in the quality of terms extracted from samples. However, common tools for natural language processing can be not suitable to deal with short texts, demanding proper tools for work in such a context [7, 18, 41].

Even after dealing with problems of polysemy and synonymy, resulting terms may not be enough to classify a message as spam or legitimate because original samples are usually very short. In such a context, some recent works recommend employing ontology models to analyze each term and find associated new terms (with the same meaning) in order to enrich original sample and acquire more features [36, 43].

[9]Lingo is an abbreviated language commonly used on mobile and Internet applications, such as SMS, *chats*, emails, blogs and social networks.

In this scenario, we have designed and evaluated a text pre-processing approach to automatically normalize and provide semantic information for noisy and short text samples in order to enhance IM and SMS spam filtering. Our hypothesis is that such processing can increase the semantic information and consequently improve learning and predictions quality. Although such a proposal was evaluated in the context of SMS spam due to the availability of data, we highlight that our technique can also be applied to deal with messages sent by online Instant Messaging apps, since they have the same text characteristics.

In order to make use of semantic information, we have designed a cascade process in which we first transcribe the original messages from its raw form into a more standardized English language, in order to allow further and more accurate text analysis. We then extract semantic relations from the lexical database BabelNet [44], and apply Word Sense Disambiguation [1], intending to make this information more accurate. Finally, we expand the original message content with the extracted information, and make use of this normalized and expanded text representation to follow a traditional machine learning approach over the messages content. According to our experiments and statistical tests, this pre-processing can improve spam filtering effectiveness.

The remainder of this paper is organized as follows: in Section 2, we briefly review the main areas of interest covered in this work. Section 3 describes the proposed expansion method. In Section 4, we describe the dataset, performance measures and main settings used in the experiments. Section 5 shows the achieved results and details the performed statistical analysis. Finally, in Section 6, we present the main conclusion and outlines for future work.

2. Related work

Our work is mainly related to three research areas:

1. The employment of natural language techniques for chat and social media lexical normalization [29];
2. Using of lexical databases and semantic dictionaries in text representation for classification [24]; and
3. The applications themselves, namely content-based Instant Messaging and SMS spam filtering [3, 25, 39, 40].

Lexical normalization is the task of replacing lexical variants of standard words and expressions normally obfuscated in noisy texts to their canonical forms, in order to allow further processing at text processing tasks. For instance, terms like "goooood" and "b4" should be replaced for the standard English words "good" and "before", respectively.

Lexical normalization is strongly related to spell checking, and in fact, many approaches in literature share techniques from this task. For instance, Cook and Stevenson [13] and Xue *et al.* [63] propose multiple simple error models, where each one captures a particular way in which lexical variants are formed, such as phonetic spelling (e.g., epik – "epic") or clipping (e.g., goin – "going").

To the best of our knowledge, the closest work to our proposal is that followed by Aw *et al.* [5], Henríquez and Hernández [30] and Kaufmann and Kalita [34], who address the problem as a machine translation task in which the goal is to statistically translate noisy text into standard English. Such works use sophisticated language models trained on noisy text samples, while our approach follows a relatively simple word-by-word translation and normalization model.

Regarding the *employment of lexical databases (LDBs) in text classification*, there is a long history of approaches working with the LDB WordNet [42] in tasks such as information retrieval [27], text categorization [24], or text clustering [32]. Other LDBs used in text classification include ConceptNet [37] and BabelNet [44], and specific LDBs or extensions used in particular tasks like Wordnet-Affect [58] and SenticNet [60] for opinion mining. For supervised tasks, there are two main approaches [24] when using a concept dictionary like WordNet and others:

- *Semantic indexing*[10]: replacing words in text documents and/or category names by their synonyms according to the concept the target word belongs to. For instance, concepts are represented in WordNet as synonym sets like, e.g., {car, auto, automobile, machine, motorcar} (a motor vehicle with four wheels) or {car, railcar, railway car, railroad car} (a wheeled vehicle adapted to the rails of railroad) for the word "car".

- *Concept indexing*: replacing (or adding) words by actual concepts in text documents. For instance, the two previous WordNet synsets have codes 02961779 and 02963378 as nouns. In consequence, any occurrence of the word "car" may be replaced by the corresponding code of the appropriate synset.

Concepts in LDBs may expand over sequences of words (collocations or multi-word expressions) and might be related to other concepts. In ConceptNet, for instance, the collocation "Ford Escort" (in which each individual word has its own meaning – appropriate concept) corresponds to the concept "ford escort". This concept is additionally related to the concept "car" by the relation "IsA" (A is a B, the hyponimy relation presented in other LDBs like WordNet as well). This relations and others make LDBs a kind of semantic networks which relations are also used in text classification (e.g., Scott & Matwin [51]). It is worth noting that some authors consider working with multi-word expressions a "semantic approach" in comparison with working with isolated words, that may be considered a "word-based" approach [10]. In our discussion between semantic and concept indexing, both the semantic

[10]This approach is named *Query Expansion* in Gómez Hidalgo *et al.* [24] because it is applied to category names, but in the general case it can be applied to any kind of text, specifically documents to be categorized.

and the word-level approach are instances of semantic indexing, unless collocations are identified by their concept codes in the semantic approach, thus leading to concept indexing.

Both in the case of concept and semantic indexing, documents must be indexed and a training process is typically applied in order to generate a classifier, by using Machine Learning algorithms such as those used in this paper. However, using LDB concepts add complexity to identifying correct meanings (or appropriate concepts) for each word occurrence, a problem that is called *Word Sense Disambiguation* (WSD). There are many approaches to WSD, as it is a popular task nearly always required in deep NLP tasks [12]. Among them, we can note that two main approaches involve using Machine Learning over a manually disambiguated text collection like SemCor [35] (supervised WSD), and using information in dictionaries (e.g., words in definitions) or in the LDB (e.g., semantic relations in WordNet) in order to define distances between concepts and use them to rank potential concepts for a word in a context [44, 45, 55] (unsupervised WSD).

In this work, we have used the LDB BabelNet [44], much more complete, recent and less used than WordNet in text classification, and we basically apply the WSD unsupervised algorithm, following the Semantic Expansion method described in Gómez Hidalgo *et al.* [24] but applied to documents instead of category names.

With respect to the tasks themselves, namely *Instant Messaging and SMS spam filtering*, many approaches borrowed from email spam filtering have been applied to it (e.g., Liu *et al.* [39] and Gómez Hidalgo *et al.* [25]). Nevertheless, the dominant approach is still content-based analysis, essentially replicating Bayesian spam filters [3, 16, 26, 56]. In these works, messages are represented in terms of the words that they contain, and Machine Learning is applied on this representation in order to induce an automated classifier that is able to infer if new messages are spam or legitimate. For instance, Cormack *et. al.* [15] study the problem of content-based spam filtering for short text messages that arise in three different contexts: SMS, blog com-ments, and email summary information such as might be displayed by a low-bandwidth client. Their main conclusions are that short messages contain an insufficient number of words to properly support bag-of-words or word bigram based spam classifiers and, as a consequence, the filter's performance is improved markedly by expanding the set of features to include orthogonal sparse word bigrams [52] and also to include character bigrams and trigrams. The same problem with message length is mentioned by Liu *et. al.* [39] at IM spam filtering when using the Bayesian email spam filter applied to classify IM messages and by Almeida *et. al.* [4] when applying several traditional machine learning techniques to filter SMS spam samples.

Other authors propose additional text representation techniques. For example, Liu and Wang [38] present an index-based online text classification approach that takes advantage of trigrams. Additionally, Maroof [40] makes use of message statistics like word length, message length, and other attributes, to discriminate between legitimate and spim messages. However, to the best of our knowledge, there is no work available in the literature that has used semantic and/or conceptual information in text representation of IM and SMS spam filtering. As an exception, and instead of basically using words as features for representing SMS messages, Sohn *et al.* [53] proposes to make use of stylistic features in message representation, while Xu *et al.* [62] make use of non-content features like time and network traffic in the same learning-based approach.

3. The proposed text expansion method

Shallow text representations like simple bag-of-words have often been shown to be limiting the performance of machine learning algorithms in text categorization problems [23]. With the goal of improving spam over Instant Messaging and mobile phone detection, this paper presents and evaluates a text pre-processing approach composed by techniques to normalize, expand and generate better text representations from noisy and short texts, in order to produce better attributes and enhance classification performance.

The expansion method combines the state-of-the-art techniques for lexical normalization and context detection, along with semantic dictionaries. In this work, each raw text sample is processed in three different stages, each one generating a new output representation in turn:

1. *Text normalization*: used to normalize and translate words in Lingo, which is the name of language commonly used on the Internet and SMS, to standard English language.
2. *Concepts generation*: used to obtain all the concepts related to a word, that is, each possible meaning of a certain word.
3. *Word sense disambiguation*: used to find the concept that is more relevant according to the context of message, among all concepts related to a certain word.

The *Concepts generation* and *Word sense disambiguation* processes are based on LDB BabelNet, which is the largest semantic repository currently available [44, 45]. While *Concepts generation* consists of replacing a given word for each of related concepts, *Word sense disambiguation* automatically selects the most relevant concept for each word. It is done through semantic analysis performed by WSD unsupervised algorithm described in Navigli & Ponzetto [45].

The proposed text pre-processing approach expands a raw text sample by first splitting it in tokens and then processing them in the described stages, generating new normalized and expanded samples[11]. This way, given a pre-defined merging rule, the expanded samples are then joined into a final output that can be processed by a machine learning method in the place of original sample. Figure 1 illustrates the process.

In the following sections, we offer more details regarding how each stage is performed.

[11]The proposed technique is publicly available at `http://lasid.sor.ufscar.br/expansion/`. We highlight that such tool is still under constant development and evaluation.

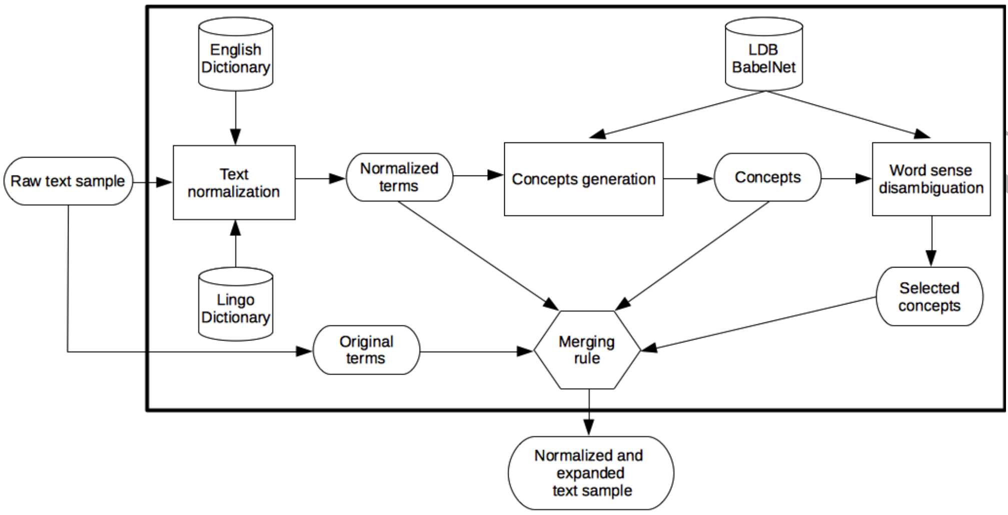


Figure 1: The original sample is processed by semantic dictionaries and context detection techniques. Each one creates a new normalized or expanded sample. Then, given a merging rule, the samples are joined into a final output represented by a text message with the same semantic content of the original sample.

3.1. Text normalization

In this stage, we have employed two dictionaries. The first is an English one used to check whether a term is an english word and then normalize it to its root form (e.g., "is" → "be" and "going" → "go"). The second is the Lingo dictionary, which is used to translate a word from Lingo to English. The process starts by looking up each word of sample in the English dictionary. In this case, our method uses the Freeling English dictionary[12]. If the word is in such a dictionary, it is then normalized to its root form. Otherwise, the word is looked up in the Lingo dictionary, which in this case is the NoSlang dictionary[13]. If the Lingo dictionary does not have a translation for the word,

[12] *Freeling English dictionary.* Available at: `http://devel.cpl.upc.edu/freeling/`.

[13] *NoSlang*: Internet Slang Dictionary & Translator. Available at: `http://www.noslang.com/dictionary/full/`.

the original is kept.

3.2. Concepts generation

The concepts are provided by LDB BabelNet repository. Since it requires English words as input, the method first employs *Text normalization* to certify that each word is indeed an English one. After that, the method removes words that belong to a stop word list, which contains articles, pronouns, prepositions, and common words[14]. The remaining words are then semantically analyzed to find their concepts.

3.3. Word sense disambiguation

Since the *concepts generation* stage can provide a huge amount of concepts for each word in the original sample, we have implemented a disambiguation step according to the algorithm proposed by Navigli and Ponzetto [45]. Basically, this algorithm looks up the most relevant concepts, according to the context of sample. First, the algorithm employs context and semantic analysis to score all concepts returned by BabelNet repository. The method then selects the concepts with the highest scores to be used instead of all possible concepts. The score is obtained by computing a number of distances in the graph constructed by the semantic network defined in BabelNet.

3.4. Merging rule

As we have the original sample along with the three above mentioned expansion stages, we have four different parameters to set for defining the merging rule, which are basically answers for the following questions, respectively: 1. *Should it keep the original tokens?*, 2. *Should it perform text normalization?*, 3. *Should it perform the concepts generation?* and 4. *Should*

[14]The list of stop words is: {a, an, are, as, at, be, by, for, from, had, has, have, he, how, i, in, is, it, of, on, or, she, that, the, they, this, to, too, was, we, were, what, when, where, who, whose, will, with, you}.

it perform the word sense disambiguation?. As each choice is binary, we have eleven possibilities of settings to expand each sample (including keeping the original samples). Note that, the output of concepts disambiguation process is always a subset of the output produced by concepts generation stage. Table 1 presents each possible set of parameters that can be used in the merging rule.

Table 1: All possible rules that can be used in the expansion method.

Rules	**Original terms**	**Normalized terms**	**Concepts**	**Selected concepts**
Rule 1	X			
Rule 2	X	X		
Rule 3	X		X	
Rule 4	X		X	X
Rule 5	X	X	X	
Rule 6	X	X	X	X
Rule 7		X		
Rule 8			X	
Rule 9			X	X
Rule 10		X	X	
Rule 11		X	X	X

In our experiments, we have performed all possible merging rules, generating one different expanded dataset for each possible set of parameters. Therefore, the original corpus (Rule 1) and ten created expanded datasets (from Rule 2 to 11) were evaluated.

3.5. An example of expansion

Table 2 presents an example of expansion achieved for a real sample. It shows the output acquired in each of the three stages for the original message "*Plz, call me bak asap... Ive gr8 news! :)*". Then, defining that, for instance, the merging rule is [*Text normalization + Word sense disambiguation*] (Rule 11), we would achieve the final expanded sample "*please call phone_call me back as soon as possible i have great big news news_program :)*", which could be used by the machine learning algorithms and possibly enhance the classification performance as it avoids common representation problems.

As shown in Table 2, the *Text normalization* replaces the slangs and abbreviations to their corresponding words in English. While the *Concepts generation* obtained all the concepts for each word in original sample, *Word sense disambiguation* stage kept only the concepts that are semantically relevant to the original sample. Finally, by using the *Final* output we intend to avoid traditional semantic problems such as polysemy and synonymy and, consequently, we aim to achieve better results when employing traditional machine learning techniques.

4. Experimental settings

To evaluate the effectiveness of the proposed expansion method, we have used the well-known SMS Spam Collection [3] which is a public dataset composed of 5,574 English, real and non-encoded messages, tagged accordingly being legitimate (ham) or spam. In such a paper, it was demonstrated that established text categorization approaches have their performance seriously degraded when they are applied to classify the original messages, since these are fairly short (limited to 160 characters) and rife with idioms, symbols and abbreviations. The same characteristics can be found in online Instant Messaging, social networks, forums, and so on.

In our experiments, we have tested all possible merging rules (Table 1), generating a different expanded dataset for each possible combination. Furthermore, we have evaluated the performance of several well-known machine

Table 2: Example of translation, normalization and expansion of a short text sample. $\mathcal{B}$ corresponds to the output of *Text normalization* stage. $\mathcal{C}$ shows all the concepts related to each word in the sample achieved in *Concepts generation* stage (excluding stop words). $\mathcal{D}$ presents the most relevant concepts selected according to the context of sample, achieved in *Disambiguation* stage. The *Final* sample is obtained from merging the outputs of *Text normalization* ($\mathcal{B}$) and *Disambiguation* ($\mathcal{D}$) (Rule 11).

Original ($\mathcal{A}$)	*Plz, call me bak asap... Ive gr8 news! :)*
Text normalization ($\mathcal{B}$)	*please call me back as soon as possible i have great news :)*
Concepts generation ($\mathcal{C}$)	*please birdsong call call_option caller caller-out claim cry margin_call outcry phone_call shout song telephone_call vociferation yell me backbone backrest binding book_binding cover dorsum rachis rear spinal_column spine vertebral_column as soon as possible i have great news news_program news_show newsworthiness tidings word :)*
Disambiguation ($\mathcal{D}$)	*please phone_call me as soon as possible i have big news_program :)*
Final Merging rule (Rule 11): $\mathcal{B} + \mathcal{D}$	*please, call phone_call me back as soon as possible i have great big news news_program :)*

learning algorithms under each generated dataset, in order to verify if the expansion method can enhance the classifiers performance. Table 3 lists the classification algorithms that were evaluated. To give credibility to the found results, we have selected a large range of methods which employ different classification strategies such as, compression, distance, trees and optimization-based algorithms. The most approaches are listed as the top-performance classification and data mining techniques currently available [61].

Briefly, the main behavior of used classifiers is described in the following.

Table 3: List of classification algorithms we have evaluated to check if the datasets generated with the proposed expansion method perform better than the original one.

Evaluated classification techniques
Bagging of Decision Trees (Bagging)
Binary Context Tree Weighting (BICTW)
Boosted C4.5 (B.C4.5)
Boosted Naïve Bayes (B.NB)
C4.5
Decomposed Context Tree Weighting (DECTW)
Improved Lempel-Ziv Algorithm (LZms)
K-Nearest Neighbors (KNN)
Lempel-Ziv 78 Algorithm (LZ78)
Linear SVM (L.SVM)
Logistic regression (Logistic)
Markov Compression (DMC)
Naïve Bayes (NB)
PART Decision List (PART)
Prediction by Partial Match (PPM)
Probabilistic Suffix Trees Compression (PST)
Sequential Minimal Optimization (SMO)

- *Bagging* creates new training sets by sampling the original data. Since sampling is performed with replacement, some samples may be repeated in each training set. These repeated samples are called bootstrap samples. The different models are then fitted using the bootstrap samples and combined by voting for classification tasks [9].

- *Binary Context Tree Weighting* (BICTW) is a simple application of the standard binary CTW algorithm over a binary representation of the sequence [59].

- *Boosting* combines the output of weak learners into a weighted sum representing the final classification of the boosted classifier [21, 25].

- *C4.5* constructs decision trees by information entropy. The algorithm iteratively performs a search for the node with the highest information entropy, dividing the set of samples in subsamples [48].

- *Decomposed Context Tree Weighting* (DECTW) uses a tree-based hierarchical decomposition of the multi-valued prediction problem into binary problems. Each of the binary problems is solved by a slight variation of the binary CTW algorithm [57].

- *Improved Lempel-Ziv Algorithm* (LZms) enhances LZ78 by retrieving more phrases during the learning phase and it provides a minimal context for the next phrase, whenever possible [46].

- *K-Nearest Neighbors* (KNN) classifies based on the class of the instances closest to it in the training space by measuring the distance between the training instances and the unknown instance [2].

- *Lempel-Ziv 78 Algorithm* (LZ78) is among the most popular lossless compression algorithms, and it was used for adult-content filtering in the past [6, 50].

- *Logistic regression* (Logistic) is a simple multinomial logistic regression model with a ridge estimator [22]. *Linear SVM* (L.SVM) is an implementation of Linear SVM that re-shapes the input features in order to apply a linear weight and scale the different features according to their predictive value [19].

- *Markov Compression* (DMC) models information with a finite state machine, the probability distribution is used to predict the next binary digit [14].

- *Naïve Bayes* (NB) is a well-known probabilistic method that applies Bayes' theorem with strong independence assumptions among the features [4].

- *PART* Decision List utilizes a divide and conquer approach to build a partial C4.5 decision tree in each iteration, making the best leaf a rule for the final set of rules [20].

- *Prediction by Partial Match* (PPM) is one of the best lossless compression algorithm and it has been used for spam filtering [8, 11].

- *Probabilistic Suffix Trees Compression* (PST) is a compression algorithm that seeks to construct the single best D-bounded Variable Markov Model according to the training sequence. It was used for the first for adult-content filtering in Santos *et al.* [50].

- *Sequential Minimal Optimization* (SMO) is a simple implementation to train Support Vector Machines. In our work, we have tested the Polynomial Kernel [47].

All evaluated methods are available in the machine learning library WEKA [28]. Even the seven compression-based models we have implemented and made them publicly available on the package `CompressionTextClassifier`[15].

In all experiments, the classifiers have been used with their default parameters, except K-nearest neighbors algorithm, in which we have employed $K = 1$, 3 and 5, and for all compression-based methods, in which we have evaluated $C = 0$ and 1. This indicates whether (1) or not (0) the adaptation of the model using the test instance is performed [6].

[15] The compression-based classifiers are available at: `http://paginaspersonales.deusto.es/isantos/files/CompressionTextClassifier/CompressionTextClassifier-0.4.3.zip`, compatible with WEKA version 3.7 or higher.

We carried out this study using the following protocol. We have used the traditional k-fold cross-validation with $k = 5$ and to tokenize the messages we have split the terms in dots, commas, tabs and spaces.

To compare the results, we have used the Matthews Correlation Coefficient (*MCC*), which is used in machine learning as a measure of the quality of binary classifications. It returns a real value between -1 and $+1$. A coefficient equals to $+1$ indicates a perfect prediction; 0, an average random prediction; and -1, an inverse prediction [4]. *MCC* provides more balanced evaluation than other measures, such as the proportion of correct predictions, especially the classes are unbalanced.

5. Results

For each evaluated classification algorithm, we have selected the merging rule in which the best performance was achieved (according to its MCC score) and called it *Expansion*. It is equivalent to perform a parameter tuning in the expansion method for each evaluated classifier. We have also selected the results attained with the original dataset, and called it *Original*.

To verify if the expanded samples can indeed enhance the classifiers performance for such application, we need to certify that results achieved with the dataset created by the proposed approach are statistically superior to the results obtained with the original dataset. Despite there are several tests that could be used to perform such analysis, the Wilcoxon Signed-Ranks Test is known to be more robust than the alternatives [17].

Such a test ranks the absolute differences in the performances of both datasets for each of the classifiers and compares the ranks for the positive and negative differences. Table 4 shows the MCC scores achieved by each classifier with the *Original* and *Expansion* databases, as well as the calculated ranks and their differences.

Then, it is necessary to calculate the indexes $R+$ and $R-$ that correspond to the sum of the ranks in which the difference is positive and negative, respectively. In our case, $R+ = 21$ and $R- = 330$.

Table 4: Ranks calculated using the Wilcoxon Signed-Ranks Test. The *Exp* column presents the results obtained using the best merging rule for each classifier; the *Orig* column shows the results for the original dataset without any pre-processing; the *Diff* column presents the difference between the results obtained with the *Original* and *Expansion* datasets, respectively; and the *Rank* column presents the ranking positions.

Classifier	**MCC**		**Diff.**	**Rank**
	Orig.	**Exp.**		
LZms C 0	0.921	0.920	0.001	2
LZms C 1	0.921	0.920	0.001	2
DMC C 0	0.939	0.938	0.001	2
PPM C 1	0.582	0.581	0.001	4
SMO	0.929	0.927	0.002	5.5
L.SVM	0.929	0.927	0.002	5.5
DECTW C 1	0.939	0.942	-0.003	7
PPM C 0	0.929	0.935	-0.006	8.5
NB	0.864	0.870	-0.006	8.5
B.C4.5	0.915	0.922	-0.007	10.5
Bagging	0.833	0.840	-0.007	10.5
B.NB	0.903	0.912	-0.009	12
PST C 0	0.800	0.810	-0.010	13
PST C 1	0.902	0.915	-0.013	14
DECTW C 0	0.781	0.797	-0.016	15
LZ78 C 0	0.876	0.894	-0.018	16.5
LZ78 C 1	0.876	0.894	-0.018	16.5
1-NN	0.771	0.800	-0.029	18
PART	0.819	0.851	-0.032	19
C4.5	0.802	0.838	-0.036	20
BICTW C 0	0.014	0.060	-0.046	21
DMC C 1	0.797	0.846	-0.049	22
Logistic	0.638	0.715	-0.077	23
3-NN	0.572	0.707	-0.135	24
5-NN	0.448	0.595	-0.147	25
BICTW C 1	-0.128	0.093	-0.221	26

Our goal is to check if the null hypothesis can be rejected, which in this case states that there is a statistical difference between the results with the expanded dataset and the original one. For the Wilcoxon Signed-Ranks Test, the null hypothesis is rejected with $\alpha = 0.05$, that is, with a confidence level of 95%, when $z \leq -1.96$. The equation for z is given by

$$z = \frac{T - \frac{1}{4}N(N+1)}{\sqrt{\frac{1}{24}N(N+1)(2N+2)}},$$

where $T = min(R+, R-)$ and N is the amount of evaluated classifiers (the same method with different parameters should also be considered).

In this case, $T = 21$ and $N = 26$, so $z = -5.55$, which means that the null hypothesis is rejected. Therefore, we can conclude that the results achieved by the classifiers using the expanded samples are statistically superior to those attained with the original ones. This means that, for such an application, the proposed text pre-processing approach can in fact provide improvements on the classifiers performance.

5.1. Parameter analysis

To find out if there is a choice of merging rule statistically superior than others for all selected classification methods, we have performed another statistical analysis under all possible expanded datasets, each one created using a different possible merging rule. However, the Friedman Test [17] indicated the null hypothesis can not be rejected. Therefore, there is no statistical difference between the results found with different merging rules.

Nevertheless, we have also analyzed if some choice of merging rule offers statistical better results for a specific set of classification algorithms. For this, we have grouped the evaluated techniques according to their classification strategies. The groups were defined as follow.

- *Compression*: BICTW, DMC, DECTW, LZ78, LZms, PPM, and PST;
- *Trees*: Bagging of trees, B.C4.5, C4.5;

- *Optimization*: Logistic, L.SVM, SMO;
- *Distance*: 1-NN, 3-NN and 5-NN;
- *Probability*: B.NB and NB.

Table 5 presents the results achieved by applying the Friedman Test under each group. As the null hypothesis can be rejected if $F_F > 6$, then for three of five analyzed groups there is a single merging rule that leads to results statistically superior than any other.

Table 5: Results achieved by statistical analysis performed on groups of classifiers using the Friedman Test. The null hypothesis is rejected if F_F is greater than the mean average rank, which in the case is 6.

Group	χ^2_F	F_F
Compression	80.61	17.65
Trees	22.93	4.03
Optimization	19.36	30.42
Distance	27.39	21.02
Probability	15.63	3.58

For groups *Compression* and *Optimization*, there are statistical evidences that the best merging rule is the combination of keeping *Original* words and those ones obtained after *text normalization*. However, for group *Distance*, the best average results were achieved by applying *text normalization* and *concepts generation*.

Despite the classifiers in groups *Trees* and *Probability* have not rejected the null hypothesis, the results have shown that some options of merging rules are, in average, better than using the original samples. In fact, trees-based classifiers performed better with terms of *text normalization* and *concepts* *generation* and probability-based methods performed better with terms attained by applying *concepts generation* and *word sense disambiguation*.

For the proposed application, such analysis demonstrate that there is not a single merging rule statistically superior for all evaluated classification approaches. Therefore, it is not possible to select *a priori* a merging rule that would fit the needs of all methods. However, once the best merging rule is found, using datasets pre-processed by the proposed expansion system clearly increase the classifiers performance if compared with the results achieved by using the original samples.

6. Conclusions and future work

The task of Instant Messaging and SMS spam filtering is still a real challenge nowadays. Two main issues make the application of established classification algorithms difficult for this specific field of research: the low number of features that can be extracted per message and the fact that messages are filled with idioms, abbreviations, and symbols.

In order to fill those gaps, we proposed a text pre-processing approach to normalize and expand short text samples in order to enhance the performance of classification techniques when applied to dealing with these short and noisy text messages. The expansion method is based on lexicography and semantic dictionaries along with the state-of-the-art techniques for semantic analysis and disambiguation. It was employed to normalize terms and create new attributes in order to change and expand the original text samples aiming to alleviate factors that can degrade performance, such as redundancies and inconsistencies.

We evaluated the proposed approach with a public, real and non-encoded dataset along with several established classification algorithms. We also performed a statistical analysis on our results, which clearly indicated that using the expansion method can effectively provide improvements on classifiers performances. Therefore, traditional filters currently in use by providers can have their performance highly increased by the employment of our proposed pre-processing technique.

Currently, we are planning to evaluate our method in applications with similar characteristics to those presented in this paper, such as content-based comment filtering and content-based spam filtering on social networks. In fact, although such a proposal was evaluated in the context of SMS spam due to the availability of data, we have enough evidence that lead us to believe that our technique can also be applied to deal with messages sent through online Instant Messaging apps (e.g., WhatsApp, Facebook Messenger, and iMessage), because they have similar text characteristics. We will also try to combine this content-based approach with traditional blacklisting techniques to enhance its filtering capabilities.

Since in our results there is no a clear winning strategy regarding feature-sets (merging rules) and classifiers, we intend to complement the traditional bag-of-words features with the ones generated by our techniques. In this way, after a feature reduction step, we will select the most relevant features to improve the classification capabilities of our method.

Regarding the expansion process, we intend to employ terms selection techniques to automatically reduce the amount of concepts brought by the LDB BabelNet repository, aiming to attenuate the noise that can in some cases be created during the concepts generation stage. Furthermore, for future work, we intend to evaluate other English semantic and lexical dictionaries, make the method able to process texts in other idioms and to evaluate the employment of ensemble classifiers to take advantage of different parameters in the same application, avoiding the requirement of identifying and selecting the best merging rule.

Acknowledgment

This work has been partially funded by the Basque Department of Education, Language policy and Culture under the project SocialSpam (PI_2014_1_62).

The authors also would like to thank the financial support of Brazilian agencies FAPESP (grant 2014/01237-7) and Capes.

References

[1] Agirre, E. & Edmonds, P. (2006). *Word sense disambiguation*. Springer.

[2] Aha, D. & Kibler, D. (1991). Instance-based learning algorithms. *Machine Learning*, **6**, 37–66.

[3] Almeida, T. A., Gómez Hidalgo, J. M., & Yamakami, A. (2011a). Contributions to the study of SMS spam filtering: new collection and results. In *Proc. of the 11th ACM DOCENG*, 259–262, Mountain View, California, USA.

[4] Almeida, T. A., Almeida, J., & Yamakami, A. (2011b). Spam Filtering: How the Dimensionality Reduction Affects the Accuracy of Naive Bayes Classifiers. *Journal of Internet Services and Applications*, **1**(3), 183–200.

[5] Aw, A., Zhang, M., Xiao, J., & Su, J. (2006). A Phrase-based Statistical Model for SMS Text Normalization. In *Proc. of the 2006 COLING/ACL*, 33–40. Association for Computational Linguistics.

[6] Begleiter, R., El-Yaniv, R., & Yona, G. (2004). On prediction using variable order Markov models. *Journal of Artificial Intelligence Research*, **22**, 385–421.

[7] Bontcheva, K., Derczynski, L., Funk, A., Greenwood, M. A., Maynard, D., & Aswani, N. (2013). Twitie: An open-source information extraction pipeline for microblog text. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, RANLP'13, 83–90, Hissar, Bulgaria.

[8] Bratko, A., Filipič, B., Cormack, G. V., Lynam, T., & Zupan, B. (2006). Spam filtering using statistical data compression models. *Journal of Machine Learning Research*, **7**, 2673–2698.

[9] Breiman, L. (1996). Bagging predictors. *Machine Learning*, **24**(2), 123–140.

[10] Cambria, E. & White, B. (2014). Jumping nlp curves: a review of natural language processing research [review article]. *Computational Intelligence Magazine, IEEE*, **9**(2), 48–57.

[11] Cleary, J. & Witten, I. (1984). Data compression using adaptive coding and partial string matching. *IEEE Transactions on Communications*, **32**(4), 396–402.

[12] Collobert, R. & Weston, J. L. (2008). A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning. In *Proceedings of the 25th International Conference on Machine Learning*, 160–167.

[13] Cook, P. & Stevenson, S. (2009). An unsupervised model for text message normalization. In *Proc. of the 2009 CALC*, 71–78. Association for Computational Linguistics.

[14] Cormack, G. V. & Horspool, R. N. S. (1987). Data compression using dynamic Markov modelling. *The Computer Journal*, **30**(6), 541–550.

[15] Cormack, G. V., Gómez Hidalgo, J. M., & Puertas Sanz, E. (2007). Spam Filtering for Short Messages. In *Proc. of the 16th ACM CIKM*, 313–320, Lisbon, Portugal.

[16] Delany, S. J., Buckley, M., & Greene, D. (2012). Sms spam filtering: Methods and data. *Expert Systems with Applications*, **39**(10), 9899–9908.

[17] Demsar, J. (2006). Statistical Comparisons of Classifiers over Multiple Data Sets. *Journal of Machine Learning Research*, **7**, 1–30.

[18] Derczynski, L., Maynard, D., Aswani, N., & Bontcheva, K. (2013). Microblog-genre noise and impact on semantic annotation accuracy. In *Proceedings of the 24th ACM Conference on Hypertext and Social Media*, HT'13, 21–30, Paris, France.

[19] Forman, G., Scholz, M., & Rajaram, S. (2009). Feature Shaping for Linear SVM Classifiers. In *Proc. of the 15th ACM SIGKDD*, 299–308, Paris, France.

[20] Frank, E. & Witten, I. H. (1998). Generating Accurate Rule Sets Without Global Optimization. In *Proc. of the 15th ICML*, 144–151, Madison, WI, USA.

[21] Freund, Y. & Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *Proc. of the 13rd ICML*, 148–156, San Francisco. Morgan Kaufmann.

[22] Friedman, J., Hastie, T., & Tibshirani, R. (1998). Additive Logistic Regression: a Statistical View of Boosting. Technical report, Stanford University, Stanford University.

[23] Gabrilovich, E. & Markovitch, S. (2007). Harnessing the Expertise of 70,000 Human Editors: Knowledge-Based Feature Generation for Text Categorization. *Journal of Machine Learning Research*, **8**, 2297–2345.

[24] Gómez Hidalgo, J. M., Buenaga Rodríguez, M., & Cortizo Pérez, J. C. (2005). The role of word sense disambiguation in automated text categorization. In *Proc. of the 10th NLDB*, 298–309, Alicante, Spain.

[25] Gómez Hidalgo, J. M., Cajigas Bringas, G., Puertas Sanz, E., & Carrero García, F. (2006). Content Based SMS Spam Filtering. In *Proc. of the 2006 ACM DOCENG*, 107–114, Amsterdam, The Netherlands.

[26] Gómez Hidalgo, J. M., Almeida, T. A., & Yamakami, A. (2012). On the Validity of a New SMS Spam Collection. In *Proc. of the 2012 IEEE ICMLA*, 240–245, Boca Raton, FL, USA.

[27] Gonzalo, J., Verdejo, F., Chugur, I., & Cigarran, J. (1998). Indexing with WordNet synsets can improve text retrieval. In *Proc. of the 1998 COLING/ACL*.

[28] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., & Witten, I. H. (2009). The WEKA Data Mining Software: An Update. *SIGKDD Explorations*, **11**(1).

[29] Han, B., Cook, P., & Baldwin, T. (2013). Lexical Normalization for Social Media Text. *ACM Trans. Intell. Syst. Technol.*, **4**(1), 1–27.

[30] Henríquez, C. & Hernández, A. (2009). A ngram-based statistical machine translation approach for text normalization on chat-speak style communications. In *Proc. of the 2009 CAW2*.

[31] Heron, S. (2009). Technologies for spam detection. *Network Security*, **2009**(1), 11–15.

[32] Hotho, A., Staab, S., & Stumme, G. (2003). Ontologies improve text document clustering. In *Proc. of the 3rd ICDM*, 541–544. IEEE.

[33] Jung, J. & Sit, E. (2004). An empirical study of spam traffic and the use of dns black lists. In *Proceedings of the ACM SIGCOMM conference on Internet measurement*, 370–375. ACM.

[34] Kaufmann, J. & Kalita, J. (2010). Syntactic Normalization of Twitter Messages. In *Proc. of the 2010 ICON*.

[35] Kilgarriff, A., England, B., & Rosenzweig, J. (2000). English Senseval: Report and Results. In *Proc. of the 2nd LREC*, 1239–1244.

[36] Kontopoulos, E., Berberidis, C., Dergiades, T., & Bassiliades, N. (2013). Ontology-based sentiment analysis of Twitter posts. *Expert Systems with Applications*, **40**(10), 4065–4074.

[37] Liu, H. & Singh, P. (2004). Conceptnet – a practical commonsense reasoning tool-kit. *BT technology journal*, **22**(4), 211–226.

[38] Liu, W. & Wang, T. (2010). Index-based Online Text Classification for SMS Spam Filtering. *Journal of Computers*, **5**(6), 844–851.

[39] Liu, Z., Lin, W., Li, N., & Lee, D. (2005). Detecting and filtering instant messaging spam – a global and personalized approach. In *Secure Network Protocols, 2005.(NPSec). 1st IEEE ICNP Workshop on*, 19–24. IEEE.

[40] Maroof, U. (2010). Analysis and detection of spim using message statistics. In *Emerging Technologies (ICET), 2010 6th International Conference on*, 246–249. IEEE.

[41] Maynard, D., Bontcheva, K., & Rout, D. (2012). Challenges in developing opinion mining tools for social media. In *Proceedings of @NLP can u tag #usergeneratedcontent?!*, LREC'12, Istanbul, Turkey.

[42] Miller, G. A. (1995). Wordnet: a lexical database for English. *Communications of the ACM*, **38**(11), 39–41.

[43] Nastase, V. & Strube, M. (2013). Transforming Wikipedia into a large scale multilingual concept network. *Artificial Intelligence*, **194**(1), 62–85.

[44] Navigli, R. & Lapata, M. (2010). An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, **32**(4), 678–692.

[45] Navigli, R. & Ponzetto, S. P. (2012). Multilingual WSD with just a few lines of code: the BabelNet API. In *Proceedings of the Association for Computational Linguistics 2012 System Demonstrations*, ACL '12, 67–72, Jeju Island, South Korea.

[46] Nisenson, M., Yariv, I., El-Yaniv, R., & Meir, R. (2003). Towards behaviometric security systems: Learning to identify a typist. In *Proc. of the PKDD'03*, 363–374. Springer.

[47] Platt, J. (1998). Fast Training of Support Vector Machines using Sequential Minimal Optimization. In B. Schoelkopf, C. Burges, & A. Smola, editors, *Advances in Kernel Methods - Support Vector Learning*. MIT Press.

[48] Quinlan, J. R. (1993). *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers Inc.

[49] Ramachandran, A., Dagon, D., & Feamster, N. (2006). Can dns-based blacklists keep up with bots? In *Proceedings of the Conference on Email and Anti-Spam (CEAS)*.

[50] Santos, I., Galán-García, P., Gómez, C. L., Nieves, J., Sanz, B., Bringas, P. G., & Gómez, J. M. (2014). An empirical study on word sense disambiguation for adult content filtering. In *International Joint Conference SOCO14-CISIS14-ICEUTE14*, 537–544. Springer International Publishing.

[51] Scott, S. & Matwin, S. (1998). Text classification using wordnet hypernyms. In *Use of WordNet in natural language processing systems: Proceedings of the conference*, 38–44.

[52] Siefkes, C., Assis, F., Chhabra, S., & Yerazunis, W. S. (2004). Combining Winnow and Orthogonal Sparse Bigrams for Incremental Spam Filtering. In *Proceedings of the 8th European Conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD '04, 410–421, New York, NY, USA.

[53] Sohn, D.-N., Lee, J.-T., Han, K.-S., & Rim, H.-C. (2012). Content-based Mobile Spam Classification Using Stylistically Motivated Features. *Pattern Recogn. Lett.*, **33**(3), 364–369.

[54] Taieb, M. A. H., Aouicha, M. B., & Hamadou, A. B. (2013). Computing semantic relatedness using wikipedia features. *Knowledge-Based Systems*, **50**(9), 260–278.

[55] Taieb, M. A. H., Aouicha, M. B., & Hamadou, A. B. (2014). A new semantic relatedness measurement using wordnet features. *Knowledge and Information Systems*, **41**(1), 467–497.

[56] Tang, G., Pei, J., & Luk, W.-S. (2014). Email mining: tasks, common techniques, and tools. *Knowledge and Information Systems*, **41**(1), 1–31.

[57] Volf, P. A. J. (2002). *Weighting techniques in data compression: Theory and algorithms*. Ph.D. thesis, Technische Universiteit Eindhoven.

[58] Weichselbraun, A., Gindl, S., & Scharl, A. (2014). Enriching semantic knowledge bases for opinion mining in big data applications. *Knowledge-Based Systems*, **69**, 78–85.

[59] Willems, F. (1998). The context-tree weighting method: Extensions. *IEEE Transactions on Information Theory*, **44**(2), 792–798.

[60] Wu, C.-E. & Tsai, R. T.-H. (2014). Using relation selection to improve value propagation in a conceptnet-based sentiment dictionary. *Knowledge-Based Systems*, **69**, 100–107.

[61] Wu, X., Kumar, V., Quinlan, J., Ghosh, J., Yang, Q., Motoda, H., McLachlan, G., Ng, A., Liu, B., Yu, P., Zhou, Z.-H., Steinbach, M., Hand, D., & Steinberg, D. (2008). Top 10 algorithms in data mining. *Knowledge and Information Systems*, **14**(1), 1–37.

[62] Xu, Q., Xiang, E., Yang, Q., Du, J., & Zhong, J. (2012). SMS Spam Detection Using Noncontent Features. *IEEE Intelligent Systems*, **27**(6), 44–51.

[63] Xue, Z., Yin, D., Davison, B. D., & Davison, B. (2011). Normalizing Microtext. In *Proc. of the 2011 AAAI*, 74–79. Association for the Advancement of Artificial Intelligence.