Noname manuscript No.
(will be inserted by the editor)

Predicting the Relevance of Social Media Posts based on Linguistic Features and Journalistic Criteria

Alexandre Pinto · Hugo Gonçalo Oliveira · Ana Oliveira Alves

Received: date / Accepted: date

Abstract An overwhelming quantity of messages is posted in social networks every minute. To make the utilization of social networks more productive, it is imperative to filter out information that is irrelevant to the general audience, such as private messages, personal opinions or well-known facts. This work is focused on the automatic classification of public social text according to its potential *relevance*, from a journalistic point of view, hopefully improving the overall experience of using a social network. Our experiments were based on a set of posts with several criteria, including the journalistic relevance, assessed by human judges. To predict the latter, we rely exclusively on linguistic features, extracted by Natural Language Processing tools, regardless the author of the message and its profile information. In our first approach, different classifiers and feature engineering methods were used to predict relevance directly from the extracted features. In a second approach, relevance was predicted indirectly, based on an ensemble of classifiers for other key criteria when defining relevance – controversy, interestingness, meaningfulness, novelty, reliability and scope – also present in the dataset. After a feature engineering step, the best results of the first approach achieved a F_1-score of 0.76 and an Area under the ROC curve (AUC) of 0.63. The best results where however achieved by the second approach, with the best learned model achieving a F_1 score of 0.84 with an AUC of 0.78. This confirmed that journalistic

A. Pinto
CISUC, Department of Informatics Engineering, University of Coimbra, Portugal
E-mail: arpinto@student.dei.uc.pt

H. Oliveira
CISUC, Department of Informatics Engineering, University of Coimbra, Portugal
E-mail: hroliv@dei.uc.pt

A. Oliveira
CISUC, Department of Informatics Engineering, University of Coimbra, Portugal
IPC, Polythecnic Institute of Coimbra, Portugal
E-mail: ana@dei.uc.pt

relevance can indeed be predicted by the combination of the selected criteria, and that linguistic features can be exploited to classify the latter.

Keywords Relevance Assessment · Social Mining · Information Extraction · Natural Language Processing · Automatic Text Classification

1 Introduction

Social networks are specifically tailored for communicating and sharing ideas. People resort to them for asking questions, sharing observations and engaging in meaningful discussions. They are also faster means of spreading and being aware of the recent news, especially when compared to traditional media like newspapers. Social networks are a huge source of information and social data, but they offer too much textual data for a single person to consume. New content is constantly published and the quantity of irrelevant information, such as private messages, personal opinions or well-known/already-known facts, grows faster than the relevant and actually useful information, causing a shrinking in valuable content. Therefore, automatic methods should be exploited to effectively identify the most relevant content (Rose, 2015; Wu, 2012) and turn searching and browsing through social networks more fruitful.

According to the Cambridge dictionary, *relevance* is "the degree to which something is related or useful to what is happening or being talked about". As a human notion, *relevance* is hard to measure and define. It is usually a goal of information retrieval methods in search engines, social media or news feeds, and is often context-dependent. For more than fifty years, the concept of *relevance* has been studied, although often implied with different terms, such as usefulness or searching, given that searching is nothing more than the retrieval of relevant answers for a given query (Saracevic, 2015).

Yet, from the journalistic point of view, where there are no clear hints or clues from the user such as query terms, search parameters, or even context, it is still hard to measure and define the relevance of a particular content. In order to simplify this task, we define relevance as a combination of six criteria: interestingness, controversy, meaningfulness, novelty, reliability and (wide) scope. Following this assumption, we exploited several linguistic features of social media posts, extracted by Natural Language Processing (NLP) tools, and machine learning approaches. First, relevance was predicted directly. In a second approach, we took a different direction and predicted each of the aforementioned criteria and only then relevance, indirectly from their combination.

Our experiments were based on a dataset of social media posts with the criteria annotated by several human judges. The second approach achieved better results – F_1 score of 0.84 and area under the ROC curve (AUC) of 0.78 – than those obtained with the first approach, even with the application of feature selection methods – F_1 score of 0.76 and AUC of 0.63.

In addition to the promising results, we see this outcome as a confirmation that relevance can be decomposed in the six considered criteria. Although the

obtained results could possibly be improved by considering additional features, such as previous data on the author of the message (e.g. followers, level of authority), or even data obtained after the publication (e.g. likes or shares), we relied exclusively on linguistic features. While, to some extent, this may seem a limitation, it enables to predict the relevance of a post even before its publication and without any previous assumption on the author, whose profile information is not always available.

The remainder of this paper starts with a brief overview on related work, focused on the automatic classification of social media contents according to similar aspects as ours. After that, we clarify our goals and describe the dataset used for experimentation and its creation. The linguistic features used are then enumerated, followed by results of using them alone, without any kind of preprocessing. After this, we report on the results obtained by combining all the extracted features and applying methods to select the most promising features. Before concluding, we report on the results of predicting relevance from an ensemble based on the six journalistic criteria. We conclude with some final remarks and cues for further work, towards the improvement of the obtained results.

2 Related Work

A number of techniques has been applied to the automatic classification of text with varied success. Here, we focus on those that target text spread in social networks.

Acknowledging the fact that users, alone, are unable to deal with so much information, automatic methods have been developed to classify social media posts according to different aspects (e.g. categories (Sriram et al, 2010), mentioned events (Ritter et al, 2012), sentiment (Nakov et al, 2013)). The previous methods can be used by content-based recommender systems (Lops et al, 2011), and may hopefully help users to organize their content and filter undesired information. Our work falls in this category, more specifically, on the prediction of the relevance of a post to a wide audience, from a journalistic point of view, which is still less trivial than the previous aspects.

On less trivial aspects of social media text, Guerini et al (2011) explored the virality of posts and defined certain phenomena in their messages, namely: virality (number of people that access it in a given time interval), appreciation (how much people like it), spreading (how much people share it), white and black buzz (how much people tend to comment it in a positive or negative mood), raising discussion (the ability to induce discussion among users) and controversy (the ability to split the audience into those that are pro and those against). The same authors developed a SVM-based classifier for automatically predicting the previous phenomena in posts of the social platform Digg, based only on the lemmas of the content words in the story and snippet. They achieved F_1 scores of 0.78 for appreciation, 0.81 for buzz, 0.70 for controversy, and 0.68 for raising-discussing.

Zeng and Wu (2013) also relied on a SVM to classify consumer reviews as helpful or not. They exploited 1 to 3-grams, together with the length of the review, its degree of detail, the given rating, and specific comparison-related keywords. With the best configuration, they reached an accuracy of 0.72.

The phenomena of popularity, surveyed by Tatar et al (2014), has some connections with virality and even relevance. Most research on the topic (e.g. Szabo and Huberman (2010)) has focused on the interactions of users (e.g. reads, appreciation, comments, shares). Yu et al (2011) used SVM and Naive Bayes classifiers to predict the popularity of social marketing messages, respectively achieving accuracies of 0.72 and 0.68, on Facebook posts by top restaurant chains that were labeled as "popular" or "not popular" according to the number of likes. The exploited features were based in simple word vectors with bag-of-words representations. They report that words such as "win", "winner", "free" or "fun" were rated as less popular, while other such as "try", "coffee", "flavors" or "new" were rated as more popular. Word ranking was based on a SVM classifier that used only boolean features representing the presences or absences of the words.

Based on the category of a news article, subjectivity of language used, mentioned named entities and the source of the publisher, R. Bandari, S. Asur and B. Huberman (2012) achieved an overall accuracy of 0.84 when predicting the number of tweets that would mention it.

Besides the number of likes, the number of times a publication is shared or mentioned in other publications is also an acceptable popularity measure. Mostly based on the features of a tweet's author (e.g. followers, favorites), Petrovic et al (2011) predicted whether it would be retweeted or not, with an F_1 score of about 0.47. L. Hong and B. Davison (2011) addressed the same problem achieving an F_1 score of 0.60. They used content features, e.g. TF-IDF scores and LDA topic distributions; topological features, e.g. PageRank scores and reciprocal links; temporal features, e.g. time differences between consecutive tweets, average time difference of consecutive messages and average time for a message to be retweeted; and meta information features, e.g. whether a message had been retweeted before or the total number of tweets produced by an user.

Fernandes et al (2015) exploited a large set of features to predict the popularity of Mashable news articles, based on the number of times they would be shared. Considered features included the length of the article, its title and its words, links, digital media content, time of publication, earlier popularity of referenced news, keywords of known popular articles, and several NLP features, such as topic, subjectivity and polarity. The best F_1 score (0.69) and accuracy (0.67) was achieved with a Random Forest classifier.

Only based on their content, tweets mentioning trending topics were classified as related or unrelated (e.g. spam) with a F_1 score of 0.79, using a C4.5 classifier, and 0.77, using Naive Bayes (Irani et al, 2010). K. Lee, D. Palsetia, R. Narayanan, Md. Ali, A. Agrawal and A. Choudhary (2011) also address the problem of assigning trending topics to categories. Using TF-IDF word vector

counts, they achieved accuracies of 0.65 and 0.61 with Naive Bayes and SVM classifiers, respectively.

The limitations of traditional bag-of-word classification models in microblogging platforms have been pointed out, due to the short length of documents Sriram et al (2010). They are thus often combined with other features, such as the author's name, presence of shortening of words and slang, time-event phrases, opinion words, emphasis on words, currency and percentage signs and user mentions at the beginning and within the post. The previous features were used to classify tweets into a set of categories (News, Events, Opinions, Deals, and Private Messages) with an accuracy of 0.95. In this case, a bag-of-words did not improve the results. In fact, the author's name seemed to be the key feature.

Frain and Wubben (2016) created a dataset for satire classification in news articles. Exploiting features like profanity, punctuation, positive and negative word counts, and bag-of-words models of unigrams and bigrams, they achieved a F_1 score of 0.89, using SVM classifiers in this task,

Although with a different goal, the previous works have focused on classifying social media posts automatically, according to some criteria, some of which (e.g. virality, helpfulness, popularity) related to our target goal, relevance, due to their ambiguous nature and dependence to an external context, not always available. This is also why this kind of classification is not straightforward and faces a challenge for automatic systems.

To our knowledge, the closest to our work is Figueira et al (2016), who also aim to detect if social network posts are relevant (roughly, news) or irrelevant (roughly, chat). They achieve an accuracy of 0.59 and a F_1 of 0.68 in a small dataset, while exploting a small set of features, namely: (a) length of a post; (b) set of words typically used in credible posts (not reported); (c) number of occurrences of certain words; (d) excessive punctuation; (e) abundant use of smileys/emoticons.

Our approach is also based on the extraction of a set of features from each post and on the exploration of a set of algorithms for learning a model that would classify the post, based on the features extracted. Besides the target classes, an important difference is that we rely exclusively on linguistically-driven features, extracted by NLP tools, which enables to predict the relevance of a post even before its publication and without any previous assumption on the author. Although with a similar goal to Figueira et al (2016), our feature set is richer, which results in a considerable improvement on the results, as it will be further reported. We should still add that, although not explicit, all their features are captured by our feature extraction approach.

3 Relevance of Social Media Posts

The proposed system has at its classifier's core a relevance filter, which classifies short texts based on their predicted relevance from a journalistic point of view. This section clarifies the adopted notion of relevance and presents

the dataset used to assess our approach, how it was created and annotated according to our criteria.

3.1 Clarifying Relevance

Social networks are people-centric and not topic-centric. As we follow more people, the likelihood of getting exposed to irrelevant content increases. Moreover, what is relevant to someone might be completely irrelevant to someone else. In fact, only a small fraction of the information spread is truly relevant to most people.

Given that relevance can be hard to measure, a discussion was conducted towards simplifying its definition. This lead to the identification of six journalistic criteria that should be considered when measuring the relevance of a piece of text, namely:

- **Interestingness**, e.g. whether it may hold the audience attention;
- **Controversy**, e.g. whether it is prone of raising discussions;
- **Meaningfulness**, e.g. whether it is valuable for the wider audience and not just to a restricted number of people;
- **Novelty**, e.g. whether the contained information is fresh or already known;
- **Reliability**, e.g. whether it sounds credible or its source it credible;
- **Scope**, e.g. whether it affects a wider audience or just a restricted number of people.

This is the definition of relevance adopted in this work. Data from social networks was collected and annotated by human volunteers, according to these criteria, plus relevance. This process is described in the following section. The resulting dataset was then used in a set of experiments towards the prediction of relevance, directly, and indirectly, from an initial prediction of the six adopted criteria, always relying on a large set of linguistic-features. This prediction is the main focus of this work and is extensively described in the following sections.

3.2 Used Dataset

The dataset used in our experiments contains textual messages gathered either from Twitter or Facebook, using their official APIs, in the period between the 16th to the 20th of April, 2016.

Tweets were retrieved with a set of search queries, related to highly discussed topics at the time, namely: (i) *refugees Syria*; (ii) *elections US*; (iii) *Olympic Games*; (iv) *terrorism Daesh*; (vi) *Referendum UK EU*. Retweets were not used.

Facebook posts and comments were gathered from the official pages of several international news websites[1]. While most of the posts in those pages would probably be relevant, comments would contain more diverse information, from this point of view. Their presence would thus ensure that relevant and irrelevant information was present. Moreover, all the posts used were written in English, had between 8 and 100 words, and did not use profanity words.

In days following their gathering, the collected posts were uploaded to the CrowdFlower[2] crowdsourcing platform, where an annotation task was launched. Given a post, volunteer contributors were asked the following questions, using a 5-point Likert scale to classify each of the six defined criteria, plus relevance: (a) interesting, in opposition to not interesting; (b) controversial or not; (c) meaningful for a general audience in opposition to private/personal; (d) new, in opposition to already known; (e) reliable or unreliable; (f) wide or narrow scope; and, finally, (g) relevant or irrelevant for a wider audience (Relevance). To ensure some degree of quality, the texts were classified by at least three different contributors, all of them with the top Crowdflower quality level (3), either from USA or UK, in order to control cultural differences.

The previous data was simplified to make our task a binary classification problem. For that purpose, the median of the answers given by the different contributors was computed and, if it was 4 or 5, the post was considered to be relevant, otherwise, irrelevant. The same method was employed for the other journalistic criteria.

The resulting dataset contains 130 Facebook posts, 343 Facebook comments and 468 tweets, in a total of 941 documents. Of those, 521 were annotated as relevant, 552 as interesting, 549 as controversial, 572 as meaningful, 391 as novel, 334 as reliable, 350 as wide scope. A number of relevant posts higher than expected is mostly explained by the trending search queries used in their retrieval. But this was needed to have a dataset that was balanced regarding the target classes. Tables 1, 2 and 3 show the proportion of posts labeled as positive for each considered criteria with each search query, respectively for each source. Table 4 focuses on the relevant posts for each query and source. As expected, the proportion of relevant posts is higher in the Facebook posts. On the other hand, it is lower in the tweets and more mixed in the Facebook comments. The search queries with the highest proportion of relevant posts were *Refugees Syria* and *Referendum UK*, although the latter included a lower number of posts.

Table 5 illustrates the contents of the dataset with four selected messages and their source. For the same messages, table 6 shows the answers of the three volunteers for each criteria and the resulting class.

[1] Source Facebook pages were: Euronews, CNN, Washington Post, Financial Times, New York Post, The New York Times, BBC News, The Telegraph, The Guardian, The Huffington Post, Der Spiegel International, Deutsche Welle News, Pravda and Fox News

[2] `https://www.crowdflower.com/`

Table 1: Facebook posts in the dataset, by query, and proportion for each criteria.

Search Word	Interesting	Controverse	Meaningful	Novel	Reliable	Wide Scope
Refugees Syria	21 (88%)	16 (67%)	19 (79%)	17 (71%)	18 (75%)	15 (63%)
Elections US	29 (59%)	10 (35%)	25 (86%)	12 (41%)	14 (48%)	15 (52%)
Olympic Games	2 (100%)	1 (50%)	2 (100%)	2 (100%)	2 (100%)	1 (50%)
Terrorism Daesh	59 (83%)	46 (65%)	61 (86%)	33 (47%)	42 (59%)	43 (61%)
Referendum UK EU	3 (75%)	1 (25%)	4 (100%)	2 (50%)	4 (100%)	3 (75%)

Table 2: Facebook comments in the dataset, by query and proportion for each criteria.

Search query	Interesting	Controverse	Meaningful	Novel	Reliable	Wide Scope
Refugees Syria	30 (70%)	38 (88%)	32 (74%)	11 (26%)	13 (30%)	21 (49%)
Elections US	21 (60%)	20 (57%)	21 (60%)	13 (37%)	12 (34%)	12 (34%)
Olympic Games	5 (100%)	4 (80%)	4 (80%)	5 (100%)	2 (40%)	4 (80%)
Terrorism Daesh	181 (72%)	203 (81%)	170 (68%)	80 (32%)	70 (28%)	97 (39%)
Referendum UK EU	6 (75%)	7 (88%)	7 (88%)	2 (25%)	1 (13%)	0 (0%)

Table 3: Tweets in the dataset, by query and proportion for each criteria.

Search query	Interesting	Controverse	Meaningful	Novel	Reliable	Wide Scope
Refugees Syria	55 (71%)	53 (68%)	56 (72%)	38 (49%)	40 (51%)	39 (50%)
Elections US	27 (40%)	32 (47%)	32 (47%)	28 (41%)	16 (24%)	16 (24%)
Olympic Games	21 (15%)	9 (7%)	36 (27%)	54 (40%)	43 (32%)	19 (14%)
Terrorism Daesh	91 (54%)	100 (59%)	88 (52%)	82 (49%)	47 (28%)	56 (33%)
Referendum UK EU	13 (72%)	9 (50%)	15 (83%)	12 (67%)	10 (56%)	9 (50%)

Table 4: Proportion of Relevant Posts in the dataset, grouped by search query and source.

Search query	Facebook posts	Facebook comments	Tweets
Refugees Syria	20 (83%)	30 (70%)	55 (71%)
Elections US	21 (72%)	21 (60%)	29 (43%)
Olympic Games	2 (100%)	4 (80%)	22 (16%)
Terrorism Daesh	55 (78%)	152 (60%)	85 (51%)
Referendum UK EU	4 (100%)	7 (88%)	14 (78%)

Content	Source	Post #
Putin: Turkey supports terrorism and stabs Russia in the back	FB post	1
Canada to accept additional 10,000 Syrian refugees	Tweet	2
Lololol winning the internet and stomping out daesh #merica	Tweet	3
Comparing numbers of people killed by terrorism with numbers killed by slipping in bath tub is stupid as eff. It totally ignores the mal-intent behind terrorism, its impact on way of life and ideology.	FB comment	4

Table 5: Examples of messages in the dataset.

4 Linguistic Features for Relevance Prediction

Our approach to the prediction of relevance consisted of learning a classification model from the dataset, based on a total of 4,579 linguistic features, extracted from the text of each post. This section presents the exploited linguistic features and describes initial results, when all the features are used to predict relevance.

Criteria	Post #	Answers			Class
		A1	A2	A3	
Controversy	1	4	5	3	Controversial
	2	3	3	4	Uncontroversial
	3	1	1	1	Uncontroversial
	4	2	5	4	Controversial
Interestingness	1	2	5	4	Interesting
	2	4	4	4	Interesting
	3	1	1	1	Uninteresting
	4	2	4	4	Interesting
Meaningfulness	1	3	4	4	Meaningful
	2	4	5	5	Meaningful
	3	2	1	1	Meaningless
	4	3	5	4	Meaningful
Novelty	1	4	2	4	Novel
	2	4	3	4	Novel
	3	2	5	1	Old
	4	2	5	3	Old
Reliability	1	4	3	3	Unreliable
	2	3	5	4	Reliable
	3	1	2	1	Unreliable
	4	4	3	3	Reliable
Scope	1	4	3	4	Wide
	2	2	5	5	Wide
	3	1	2	1	Narrow
	4	2	4	3	Narrow
Relevance	1	5	4	5	Relevant
	2	4	5	5	Relevant
	3	1	1	1	Irrelevant
	4	2	4	3	Irrelevant

Table 6: Answers on their labels and the resulting class

4.1 Feature Extraction

The full set of exploited features covers a wide range of linguistic information, such as the counts of different part-of-speech (PoS), chunk and named entity (NE) tags, plus one feature for each different tag of the previous, as well as sentiment features, namely the number of positive and negative words in the message. Those were used together with topic modelling features, namely the LDA distribution in 20 topics, and n-gram features, namely 1 to 3-grams of tokens (with frequencies of 3 or more), the 750 most frequent 1 to 5-grams of lemmas and stems, and the 125 most frequent 1 to 5-grams of PoS and chunk tags. Our assumption is that a combination of the previous features will roughly capture some of the aspects that make text fall in the relevant or in the irrelevant class. For instance, relevant news text should transmit an interesting event and thus answer the questions of *who* or *which*, *when* and *where*, often named entities, and *why*, communicated through certain linguistic patterns, hopefully captured by the n-gram features. News text should also be reliable and unbiased, not appealing to sentiment, and be written in a more formal level than random irrelevant comments, which should be captured by

the presence of certain tokens or n-grams. Table 7 details the feature set and the distinct number of features of each kind.

Feature Set	**#Distinct Features**
PoS-tags	54
Chunk tags	23
NE tags	11
Total number of PoS/Chunk/NE tags	3
Total number of positive/neutral/negative words	3
Total number of characters/tokens	2
Total number/proportion of all capitalized words	2
LDA topic distribution	20
Token 1-3grams	2711 ($freq \geq 3$)
Lemma 1-5grams	top-750 ($freq \geq 1$)
Stem 1-5grams	top-750 ($freq \geq 1$)
PoS 1-5grams (1-5)	top-125 ($freq \geq 1$)
Chunk 1-5grams (1-5)	top-125 ($freq \geq 1$)
Total	4,579

Table 7: Feature sets used

Features were extracted with several tools and resources available. Since we were working with social media text, TweetNLP (K. Gimpel, N. Schneider, B. O'Connor, D. Das, D. Mills, J. Eisenstein, M. Heilman, D. Yogatama, J. Flanigan and N. Smith, 2011) was used for PoS tagging, and Twitter NLP (Ritter et al, 2011) for chunk tagging and named entity recognition, both using the pre-trained available models. NLTK (S. Bird, 2006) was used for lemmatisation and stemming. The word sentiment classification was based on Hu and Liu's Opinion Lexicon[3], a manually created list of English words and their associated polarity. Words not present were considered to be neutral. Scikit-learn[4] (Pedregosa et al, 2011) was used for extracting n-grams, and the Python LDA module[5] for computing topic distributions.

4.2 Initial Results

In our initial experiments, the extracted features were used for predicting relevance, all together, and also grouped in smaller sets of related features. No preprocessing or feature engineering methods were applied.

Classification experiments were performed with the Scikit-learn machine learning toolkit. Different classification methods – Minimum Distance (MinDist) and k-Nearest Neighbours (kNN), both based on the Euclidean distance, Naive Bayes, Support Vector Machines (SVM), Decision Tree and Random Forest (RF) – were tested with each feature set, in a 10-fold cross validation, which enabled the computation of the average and standard

[3] `https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html\#lexicon`

[4] `http://scikit-learn.org/stable`

[5] `https://pypi.python.org/pypi/lda`

deviation for common evaluation metrics – Accuracy, Precision, Recall, F_1 score, Area under the Precision-Recall curve (AP), and area under the ROC curve (AUC). Tables 8, 9 and 10 show the previous results, respectively for the MinDist, SVM and RF-based classifiers. Although differences where not substantial to the other classifiers, those achieved the highest results.

Table 8: Initial Results for the Minimum Distance Classifier

Metrics / Feature Set	Accuracy	Precision	Recall	F_1	AP	AUC
All features	0.57 ± 0.05	**0.72 ± 0.14**	0.41 ± 0.14	0.50 ± 0.11	0.73 ± 0.07	0.59 ± 0.05
PoS tags	0.57 ± 0.06	0.71 ± 0.15	0.42 ± 0.14	0.50 ± 0.11	0.72 ± 0.07	0.58 ± 0.06
Chunk tags	0.57 ± 0.05	0.71 ± 0.14	0.42 ± 0.13	0.51 ± 0.10	0.72 ± 0.07	0.59 ± 0.05
Named entities	0.57 ± 0.05	0.71 ± 0.14	0.42 ± 0.13	0.51 ± 0.10	0.72 ± 0.07	0.58 ± 0.05
Chars,Tokens,Caps	0.57 ± 0.06	0.71 ± 0.14	0.41 ± 0.14	0.50 ± 0.11	0.73 ± 0.07	0.59 ± 0.05
Sentiment	0.56 ± 0.05	0.70 ± 0.15	0.41 ± 0.13	0.50 ± 0.11	0.72 ± 0.07	0.58 ± 0.05
LDA	**0.63 ± 0.15**	0.65 ± 0.17	**0.89 ± 0.17**	**0.73 ± 0.12**	**0.80 ± 0.09**	**0.60 ± 0.17**
Token n-grams	0.57 ± 0.06	0.70 ± 0.16	0.46 ± 0.15	0.53 ± 0.11	0.73 ± 0.08	0.58 ± 0.06
Lemma n-grams	0.58 ± 0.05	0.71 ± 0.14	0.48 ± 0.15	0.55 ± 0.10	0.74 ± 0.07	**0.60 ± 0.05**
Stem n-grams	0.59 ± 0.05	**0.72 ± 0.13**	0.48 ± 0.15	0.55 ± 0.10	0.74 ± 0.06	**0.60 ± 0.05**
PoS n-grams	0.58 ± 0.05	0.71 ± 0.11	0.46 ± 0.16	0.53 ± 0.12	0.73 ± 0.05	0.59 ± 0.05
Chunk n-grams	0.57 ± 0.06	0.70 ± 0.15	0.43 ± 0.14	0.51 ± 0.11	0.72 ± 0.07	0.59 ± 0.05

Table 9: Initial Results for the Support Vector Machine Classifier

Metrics / Feature Set	Accuracy	Precision	Recall	F_1	AP	AUC
All features	0.60 ± 0.10	**0.67 ± 0.14**	0.63 ± 0.08	0.64 ± 0.08	0.75 ± 0.07	0.60 ± 0.11
PoS tags	0.63 ± 0.08	0.66 ± 0.09	0.71 ± 0.21	0.66 ± 0.12	0.76 ± 0.06	**0.62 ± 0.08**
Chunk tags	0.58 ± 0.07	0.59 ± 0.05	0.75 ± 0.24	0.64 ± 0.15	0.64 ± 0.15	0.56 ± 0.06
Named entities	0.55 ± 0.09	0.51 ± 0.29	0.64 ± 0.42	0.51 ± 0.32	0.78 ± 0.03	0.54 ± 0.07
Chars,Tokens,Caps	0.54 ± 0.06	0.45 ± 0.23	0.74 ± 0.38	0.56 ± 0.28	0.77 ± 0.02	0.51 ± 0.04
Sentiment	0.52 ± 0.05	0.59 ± 0.26	0.60 ± 0.40	0.48 ± 0.29	0.71 ± 0.15	0.51 ± 0.02
LDA	**0.64 ± 0.14**	0.65 ± 0.16	**0.85 ± 0.19**	**0.72 ± 0.13**	**0.79 ± 0.09**	0.61 ± 0.16
Token n-grams	0.60 ± 0.09	0.65 ± 0.11	0.62 ± 0.11	0.63 ± 0.09	0.74 ± 0.06	0.59 ± 0.09
Lemma n-grams	0.59 ± 0.07	0.65 ± 0.11	0.61 ± 0.08	0.62 ± 0.06	0.74 ± 0.05	0.58 ± 0.08
Stem n-grams	0.61 ± 0.09	0.69 ± 0.14	0.62 ± 0.11	0.64 ± 0.08	0.76 ± 0.07	0.61 ± 0.09
PoS n-grams	0.58 ± 0.09	0.63 ± 0.12	0.63 ± 0.15	0.62 ± 0.11	0.73 ± 0.08	0.58 ± 0.10
Chunk n-grams	0.56 ± 0.06	0.59 ± 0.05	0.69 ± 0.17	0.62 ± 0.09	0.72 ± 0.05	0.54 ± 0.05

Table 10: Initial Results for the Random Forest Classifier

Metrics / Feature Set	Accuracy	Precision	Recall	F_1	AP	AUC
All features	**0.64 ± 0.06**	0.66 ± 0.09	**0.78 ± 0.10**	**0.71 ± 0.04**	0.78 ± 0.04	**0.62 ± 0.09**
PoS tags	0.63 ± 0.07	0.66 ± 0.09	**0.78 ± 0.11**	0.70 ± 0.05	0.78 ± 0.04	**0.62 ± 0.08**
Chunk tags	0.60 ± 0.04	0.62 ± 0.04	0.70 ± 0.07	0.66 ± 0.04	0.74 ± 0.02	0.58 ± 0.04
Named entities	0.58 ± 0.06	0.62 ± 0.06	0.66 ± 0.07	0.64 ± 0.05	0.73 ± 0.04	0.56 ± 0.06
Chars,Tokens,Caps	0.55 ± 0.03	0.59 ± 0.03	0.61 ± 0.09	0.60 ± 0.05	0.71 ± 0.03	0.54 ± 0.03
Sentiment	0.52 ± 0.04	0.57 ± 0.03	0.57 ± 0.09	0.57 ± 0.05	0.69 ± 0.03	0.51 ± 0.04
LDA	0.60 ± 0.11	0.65 ± 0.16	0.75 ± 0.15	0.67 ± 0.10	0.77 ± 0.08	0.58 ± 0.13
Token n-grams	0.62 ± 0.10	0.66 ± 0.15	0.77 ± 0.16	0.69 ± 0.09	0.78 ± 0.07	0.60 ± 0.12
Lemma n-grams	0.63 ± 0.11	0.66 ± 0.16	**0.78 ± 0.13**	0.70 ± 0.09	0.78 ± 0.08	0.61 ± 0.13
Stem n-grams	0.64 ± 0.11	**0.67 ± 0.15**	0.78 ± 0.16	0.70 ± 0.10	**0.79 ± 0.08**	**0.62 ± 0.12**
PoS n-grams	0.62 +- 0.06	0.65 +- 0.10	0.77 +- 0.11	0.69 +- 0.04	0.77 +- 0.04	0.61 +- 0.08
Chunk n-grams	0.59 ± 0.04	0.59 ± 0.04	0.72 ± 0.10	0.66 ± 0.05	0.74 ± 0.03	0.57 ± 0.05

Considering the full feature set, MinDist achieved the best precision (0.72) while RF got the best accuracy (0.64), recall (0.78), F_1 (0.71), AUC (0.78) and AP (0.62). Although using a different dataset, these numbers already show better results than Figueira et al (2016), who achieved an accuracy of 0.59 and F_1 of 0.68.

Looking at the different feature sets, LDA lead to the best results. Using a simple classifier as MinDist, accuracy was 0.63 and F_1 0.73, which outperforms the same metric with the full feature set. With an SVM classifier and only LDA features, the accuracy of the full feature set (0.63) is also matched.

The results of using just PoS tags are also worth noticing, especially with the SVM and the RF classifiers. For instance, using only PoS tag features and an RF classifier, accuracy reaches 0.63, F_1 0.70, AP 0.78 and AUC 0.62, which is very close to using the complete feature set. Using lemma and stem n-gram features also lead to results very close to using the complete feature set.

These results can be seen as a baseline and were later improved by applying feature engineering methods, in order to preprocess and reduce the number of features based on their discriminating power.

5 Feature Engineering for Improving Relevance Detection

Of course, not all of the 4,579 distinct features have enough discriminative power and are not informative enough for the classification task. As shown in the previous section, some of them seem to add so much noise that the learning algorithms have trouble dealing with, otherwise the best results would have been obtained by the complete feature set. Therefore, after the experiments with different feature sets, feature engineering methods were applied to improve the quality of the final predictions.

Successful removal of noisy and redundant data typically improves the overall classification accuracy of the resulting model, reducing also overfitting. For this purpose, different methods available in the Scikit-learn tool were used for preprocessing – Standardization (Std), Normalization (Norm) and Scaling (0,1) – , for feature selection – Information Gain, Gain Ratio, Chi-Square (χ^2), Fisher Score and Pearson correlation – , as well as a feature reduction technique – Principal Component Analysis (PCA). The PCA method was applied to 4 dimensions and the Pearson correlations were used to exclude features that did not have a correlation of at least 0.2 with the target class. The classifiers were used with the same parameterization of the baseline experiments. A 10-fold cross validation was performed as well, with the feature engineering methods ran within each fold, using only the training set and no information from the test set, such as labels, and thus avoiding bias. In order to choose the number of features f, we followed a simple heuristic. It was based on the χ^2 statistic test to find the number of statistically significant features i.e., features that had a statistic score of $\chi^2 > 10.83$, which are very likely to be dependent from the target class, with only a 0.001 chance of not being so. The resulting number was $f = 210$, which is the target number of features to

select. Tables 11, 12, 13, 14 and 15 show the results of a 10-fold cross validation when predicting relevance in our dataset, with the previous methods applied to the full feature set, respectively with a MinDist, kNN, Naive Bayes, SVM, and RF classifier.

Table 11: Different Preprocessing and Feature Selection methods with a Minimum Distance Classifier

Metrics / Pipeline	Accuracy	Precision	Recall	F_1	AP	AUC
Std, full set	0.57 ± 0.05	**0.72 ± 0.14**	0.41 ± 0.14	0.50 ± 0.11	0.73 ± 0.07	0.59 ± 0.05
Norm, full set	0.57 ± 0.05	**0.72 ± 0.14**	0.41 ± 0.14	0.50 ± 0.11	0.73 ± 0.07	0.59 ± 0.05
Scaling, full set	0.57 ± 0.05	**0.72 ± 0.14**	0.41 ± 0.14	0.50 ± 0.11	0.73 ± 0.07	0.59 ± 0.05
Std, Info Gain	0.57 ± 0.05	**0.72 ± 0.14**	0.41 ± 0.14	0.50 ± 0.11	0.73 ± 0.07	0.59 ± 0.05
Std, Gain Ratio	0.57 ± 0.05	0.58 ± 0.06	**0.86 ± 0.17**	**0.68 ± 0.05**	0.76 ± 0.04	0.53 ± 0.07
Std, χ^2	0.57 ± 0.05	**0.72 ± 0.14**	0.41 ± 0.14	0.50 ± 0.11	0.73 ± 0.07	0.59 ± 0.05
Std, Fisher Score	0.51 ± 0.07	0.62 ± 0.17	0.32 ± 0.13	0.41 ± 0.11	0.66 ± 0.09	0.53 ± 0.08
Std, Pearson	**0.65 ± 0.16**	0.70 ± 0.15	0.63 ± 0.20	0.65 ± 0.18	**0.77 ± 0.11**	**0.62 ± 0.16**
Std, Pearson, PCA	**0.65 ± 0.16**	0.70 ± 0.15	0.63 ± 0.20	0.65 ± 0.18	**0.77 ± 0.11**	**0.62 ± 0.16**
Std, Gain Ratio, PCA	0.57 ± 0.07	0.55 ± 0.08	**0.86 ± 0.29**	0.65 ± 0.19	0.75 ± 0.10	0.53 ± 0.06

Table 12: Results of applying different Preprocessing and Feature Selection methods with a k-Nearest Neighbors

Metrics / Pipeline	Accuracy	Precision	Recall	F_1	AP	AUC
Std, full set	0.57 ± 0.05	0.60 ± 0.05	0.73 ± 0.12	0.65 ± 0.05	0.74 ± 0.03	0.55 ± 0.06
Norm, full set	0.57 ± 0.05	0.60 ± 0.05	0.73 ± 0.12	0.65 ± 0.05	0.74 ± 0.03	0.55 ± 0.06
Scaling, full set	0.57 ± 0.05	0.60 ± 0.05	0.73 ± 0.12	0.65 ± 0.05	0.74 ± 0.03	0.55 ± 0.06
Std, Info Gain	0.58 ± 0.05	0.61 ± 0.05	0.75 ± 0.11	0.66 ± 0.04	0.75 ± 0.03	0.56 ± 0.05
Std, Gain Ratio	0.59 ± 0.06	0.59 ± 0.06	**0.95 ± 0.11**	0.72 ± 0.04	0.78 ± 0.03	0.55 ± 0.08
Std, χ^2	0.57 ± 0.04	0.60 ± 0.05	0.75 ± 0.11	0.66 ± 0.04	0.74 ± 0.03	0.55 ± 0.05
Std, Fisher score	0.48 ± 0.06	0.58 ± 0.14	0.25 ± 0.16	0.33 ± 0.14	0.62 ± 0.09	0.51 ± 0.05
Std, Pearson	0.62 ± 0.13	0.62 ± 0.13	0.87 ± 0.12	0.72 ± 0.10	0.78 ± 0.08	0.59 ± 0.14
Std, Pearson, PCA	**0.65 ± 0.15**	**0.67 ± 0.17**	0.85 ± 0.11	**0.74 ± 0.11**	**0.80 ± 0.09**	**0.63 ± 0.16**
Std, Gain Ratio, PCA	0.54 ± 0.06	0.58 ± 0.05	0.66 ± 0.10	0.61 ± 0.06	0.71 ± 0.04	0.53 ± 0.06

Table 13: Results of applying different Preprocessing and Feature Selection methods with a Naive Bayes

Metrics / Pipeline	Accuracy	Precision	Recall	F_1	AP	AUC
Std, full set	0.60 ± 0.11	0.63 ± 0.11	0.73 ± 0.16	0.66 ± 0.12	0.75 ± 0.08	0.59 ± 0.12
Norm, full set	0.60 ± 0.11	0.63 ± 0.11	0.73 ± 0.16	0.66 ± 0.12	0.75 ± 0.08	0.59 ± 0.12
Scaling, full set	0.60 ± 0.11	0.63 ± 0.11	0.73 ± 0.16	0.66 ± 0.12	0.75 ± 0.08	0.59 ± 0.12
Std, Info Gain	0.55 ± 0.08	0.67 ± 0.14	0.41 ± 0.17	0.48 ± 0.14	0.71 ± 0.08	0.57 ± 0.08
Std, Gain Ratio	0.53 ± 0.06	0.57 ± 0.07	0.63 ± 0.11	0.59 ± 0.06	0.70 ± 0.04	0.51 ± 0.07
Std, χ^2	0.55 ± 0.06	0.67 ± 0.14	0.43 ± 0.15	0.50 ± 0.11	0.71 ± 0.07	0.57 ± 0.06
Std, Fisher score	0.50 ± 0.07	0.61 ± 0.16	0.27 ± 0.09	0.37 ± 0.11	0.64 ± 0.09	0.52 ± 0.07
Std, Pearson	**0.65 ± 0.16**	**0.66 ± 0.17**	**0.94 ± 0.11**	**0.76 ± 0.10**	**0.82 ± 0.09**	**0.61 ± 0.18**
Std, Pearson, PCA	**0.65 ± 0.16**	**0.66 ± 0.17**	**0.94 ± 0.11**	**0.76 ± 0.10**	0.81 ± 0.09	**0.61 ± 0.18**
Std, Gain Ratio, PCA	0.58 ± 0.05	0.58 ± 0.03	0.95 ± 0.10	0.95 ± 0.10	0.78 ± 0.02	0.54 ± 0.05

On their own, preprocessing methods do not lead to clear improvements in the classification of relevance. But better results are obtained when the previous are combined with the Pearson correlation filter and PCA. This leads

Table 14: Results of applying different Preprocessing and Feature Selection methods with a Support Vector Machine

Metrics / Pipeline	Accuracy	Precision	Recall	F_1	AP	AUC
Std, full set	0.58 ± 0.12	0.65 ± 0.14	0.57 ± 0.21	0.58 ± 0.17	0.73 ± 0.10	0.58 ± 0.12
Norm, full set	0.58 ± 0.09	0.66 ± 0.13	0.58 ± 0.16	0.60 ± 0.10	0.74 ± 0.07	0.59 ± 0.10
Scaling, full set	0.59 ± 0.11	0.65 ± 0.15	0.61 ± 0.19	0.61 ± 0.14	0.74 ± 0.09	0.59 ± 0.11
Std, Info Gain	0.54 ± 0.09	0.62 ± 0.14	0.53 ± 0.37	0.47 ± 0.28	0.71 ± 0.10	0.54 ± 0.07
Std, Gain Ratio	0.54 ± 0.06	0.57 ± 0.06	0.66 ± 0.09	0.61 ± 0.05	0.71 ± 0.04	0.52 ± 0.06
Std, χ^2	0.56 ± 0.11	0.61 ± 0.20	0.58 ± 0.34	0.54 ± 0.24	0.71 ± 0.14	0.56 ± 0.11
Std, Fisher score	0.54 ± 0.09	0.57 ± 0.09	0.54 ± 0.29	0.53 ± 0.18	0.68 ± 0.10	0.54 ± 0.08
Std, Pearson	**0.65 ± 0.16**	**0.66 ± 0.17**	0.94 ± 0.11	**0.76 ± 0.10**	**0.81 ± 0.08**	**0.61 ± 0.18**
Std, Pearson, PCA	0.64 ± 0.16	0.65 ± 0.17	0.94 ± 0.11	0.75 ± 0.10	**0.81 ± 0.09**	**0.61 ± 0.18**
Std, Gain Ratio, PCA	0.59 ± 0.05	0.58 ± 0.04	**0.96 ± 0.10**	0.72 ± 0.04	0.78 ± 0.03	0.54 ± 0.06

Table 15: Results of applying different Preprocessing and Feature Selection methods with a Random Forest

Metrics / Pipeline	Accuracy	Precision	Recall	F_1	AP	AUC
Std, full set	0.63 ± 0.09	0.67 ± 0.15	**0.80 ± 0.18**	0.70 ± 0.09	**0.79 ± 0.07**	0.61 ± 0.10
Norm, full set	**0.64 ± 0.11**	0.66 ± 0.16	**0.80 ± 0.19**	0.70 ± 0.12	**0.79 ± 0.09**	0.62 ± 0.13
Scaling, full set	**0.64 ± 0.11**	**0.68 ± 0.16**	**0.80 ± 0.18**	**0.71 ± 0.12**	**0.79 ± 0.08**	**0.63 ± 0.12**
Std, Info Gain	**0.64 ± 0.10**	**0.68 ± 0.15**	0.78 ± 0.13	**0.71 ± 0.08**	**0.79 ± 0.07**	0.62 ± 0.12
Std, Gain Ratio	0.53 ± 0.06	0.57 ± 0.06	0.67 ± 0.08	0.61 ± 0.05	0.71 ± 0.03	0.51 ± 0.07
Std, χ^2	0.62 ± 0.10	0.67 ± 0.16	0.76 ± 0.16	0.69 ± 0.09	0.78 ± 0.08	0.61 ± 0.11
Std, Fisher score	0.53 ± 0.11	0.56 ± 0.10	0.53 ± 0.30	0.52 ± 0.19	0.67 ± 0.11	0.53 ± 0.09
Std, Pearson	0.59 ± 0.18	0.64 ± 0.18	0.72 ± 0.13	0.67 ± 0.15	0.75 ± 0.11	0.58 ± 0.19
Std, Pearson, PCA	0.60 ± 0.18	0.64 ± 0.18	0.70 ± 0.13	0.66 ± 0.15	0.75 ± 0.12	0.59 ± 0.19
Std, Gain Ratio, PCA	0.53 ± 0.05	0.58 ± 0.06	0.62 ± 0.09	0.59 ± 0.04	0.70 ± 0.03	0.52 ± 0.06

to the best accuracy (0.65) with the MinDist, kNN, Naive Bayes and SVM classifiers; to the best F_1 (0.76) with the Naive Bayes and SVM classifiers; and to the best AUC (0.63), with the kNN classifier. The best AP (0.82) is obtained when Standardization is combined with Pearson correlation and the Naive Bayes classifier is used. In general, the RF classifier gets lower results, except for AUC, as it matches the best result (0.63) when Scaling is applied for preprocessing.

We can say that preprocessing the full feature set with the Standardization method, then selecting the features according to the Pearson correlation, and, possibly, applying PCA, would be the best choice for handling our original feature set and improving our results. Although a narrower choice could be made by looking at specific metrics, once the previous methods are used, either a MinDist, a kNN, a Naive Bayes or an SVM classifier would do an interesting job.

These set the best results obtained for predicting relevance directly, based on linguistic features. Next section goes further as relevance is decomposed in the six criteria adopted to define it, and predicted indirectly from an initial prediction of those criteria, still relying on the same features and methods.

6 Predicting Relevance from Journalistic Criteria

We recall once again that the adopted definition of relevance is based on six journalistic criteria – controversy, interestingness, meaningfulness, novelty,

reliability and (wide) scope. If our assumption is correct, this means that it should be possible to assess relevance from a combination of those criteria. Although our dataset had been created with these criteria on mind, and that each post was also manually classified according to them (see section 3.2), they had not been exploited to this point.

In order to confirm that our definition of relevance makes sense and, hopefully, to further improve our classification results, we decided to exploit the six journalistic criteria in our classification task. Those criteria would be predicted first, still relying on linguistic features, and would then be combined to predict relevance indirectly. This was done with an intermediate layer of six classifiers, each for predicting a single criteria, following a similar approach as the one used for predicting relevance directly, i.e. each classifier of this layer has textual data as input and outputs a single prediction regarding its target criteria. After this, a final classifier gets a binary prediction of each criteria as input and outputs a prediction of relevance. Figure 1 illustrates this approach.

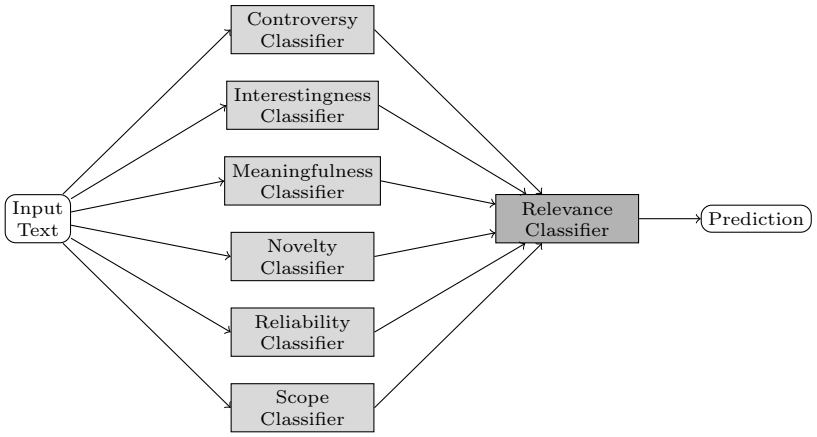

Fig. 1: Indirect prediction of relevance, from six journalistic criteria

6.1 Relevance from human judgements on journalistic criteria

To confirm that such an approach made sense, a relevance classifier was first trained from the manual annotations in dataset, regarding each of the six journalistic criteria. Refer to tables 1, 2 and 3 for the proportion of positive posts for each criteria.

Table 16 shows the results of a 10-fold cross validation with this kind of relevance classifier, with Standardization applied for preprocessing, and using different classification methods. For such a small feature set (6), no feature engineering was needed. All the remaining parameters were the same as in the initial experiments.

Table 16: Results on predicting relevance from human-annotated journalistic criteria.

Metrics / Classifier	Accuracy	Precision	Recall	F_1	AP	AUC
Min Distance	0.80 ± 0.06	**0.84 ± 0.07**	0.80 ± 0.08	0.82 ± 0.06	0.87 ± 0.04	0.80 ± 0.06
k-NN	**0.82 ± 0.07**	0.82 ± 0.07	**0.86 ± 0.08**	**0.84 ± 0.06**	**0.88 ± 0.04**	**0.81 ± 0.07**
Naive Bayes	0.81 ± 0.06	0.84 ± 0.07	0.81 ± 0.08	0.82 ± 0.06	**0.88 ± 0.04**	**0.81 ± 0.06**
SVM	0.80 ± 0.06	0.82 ± 0.07	0.84 ± 0.08	0.83 ± 0.06	0.87 ± 0.04	0.80 ± 0.07
Random Forest	0.80 ± 0.07	0.81 ± 0.07	0.84 ± 0.07	0.82 ± 0.06	0.87 ± 0.04	0.79 ± 0.07

In general, the results were positive and substantially higher than those obtained in the previous experiments. The kNN classifier achieved the best performance overall, including the best accuracy (0.82), recall (0.86), F_1 score (0.84), AP (0.88) and AUC (0.81). The MinDist Classifier obtained the best overall precision (0.84). The Naive Bayes also matched the best AP and AUC.

This not only confirms that, to some extent, journalistic relevance can be predicted from the six adopted criteria, but also that, this way, when compared to the previous experiments, the results of predicting relevance are clearly improved. Yet, we recall that these results rely on the manual annotation of each of the six journalistic criteria. In order to have a completely automatic pipeline, the journalistic criteria had also to be predicted automatically. This is described in the following section.

More than confirming our assumption on the decomposition of relevance, this experiment supported our choice of kNN as the final relevance classifier of our two-layer model. It achieved a decent area under the ROC curve (0.81), showing that even for low false positive rates such as 0.2, on average, it obtains interesting rates of true positive (> 0.7). Furthermore, in some cases, kNN outperformed the AUC mark of 0.85. Regarding the precision-recall curves, kNN achieved an area of 0.88, on average, providing a good flexibility with the trade-offs. For high values of recall, it also obtained similarly high values of precision.

6.2 Relevance from automatic predictions of journalistic criteria

In order to have a completely automatic relevance classifier, based on the prediction of the six journalistic criteria, the prediction of those criteria had also to be made automatically. For this purpose, we performed several experiments where the intermediate classifiers would predict if the post was controverse, interesting, meaningful, novel, reliable and had a wide scope. Each of the six predictions was made from exactly the same linguistic features as those exploited in the previous experiments (see table 7), with a feature engineering pipeline that achieved the best F_1 score (Standardization and Pearson correlation, see section 5), but with different classifiers. The prediction of each criteria was then used as the input of a kNN classifier, selected after the experiments of the previous section. Table 17 details the results obtained with different classification methods in the intermediate classifiers.

Table 17: Results on predicting relevance by an ensemble of predicted journalistic criteria.

Metrics / Intermediate	Accuracy	Precision	Recall	F_1	AP	AUC
Min Distance	0.62 ± 0.11	0.66 ± 0.17	0.89 ± 0.19	0.72 ± 0.08	0.80 ± 0.06	0.59 ± 0.13
k-NN	0.54 ± 0.08	0.63 ± 0.14	0.57 ± 0.17	0.57 ± 0.08	0.72 ± 0.05	0.54 ± 0.09
Naive Bayes	0.56 ± 0.01	0.56 ± 0.01	**0.97 ± 0.03**	0.71 ± 0.01	0.77 ± 0.01	0.52 ± 0.01
SVM	0.54 ± 0.03	0.56 ± 0.02	0.89 ± 0.08	0.68 ± 0.03	0.75 ± 0.02	0.50 ± 0.04
Random Forest	**0.79 ± 0.07**	**0.80 ± 0.08**	0.84 ± 0.07	**0.82 ± 0.06**	**0.86 ± 0.04**	**0.78 ± 0.08**

Random forests revealed to be the best option for the intermediate classifiers. When used to predict each of the six criteria, they achieved the best accuracy (0.79), precision (0.80), F_1 (0.82) score, AP (0.86) and AUC (0.78). The Naive Bayes obtained the highest recall (0.97).

Though lower than when using the manual annotations, the general results are substantially higher than the direct prediction of relevance, Accuracy improved 15 points, precision and recall improved 8, F_1 9 points, AP 6, and AUC 16 points.

This suggests that using a meta classifier with an intermediate layer of the best single models is better than predicting relevance directly. In this case, Random Forest classifiers were used for the intermediate layer and a kNN classifier for computing the final relevance prediction. Tackling different aspects of the problem space (different criteria) was helpful to reduce the generalization error and variance and proved to be the best option.

7 Conclusions

Several experiments were described on the automatic classification of social media posts according to their journalistic relevance, relying exclusively on linguistic features extracted from text. A decomposition of journalistic relevance in six key criteria was proposed – controversy, interestingness, meaningfulness, novelty, reliability and scope – and a dataset was created with social media posts annotated according to them, plus relevance.

Experiments involved the extraction of a large set of linguistic features, then used as the input of a classifier, trained and tested with the created dataset. We presented the results of predicting relevance with different classifiers that exploited all the extracted features, only certain groups of features, and also with the application of feature engineering methods. In a 10-fold cross validation, the best results of this approach achieved an accuracy of 0.65 and a F_1 score of 0.76, with a Naive Bayes or a SVM classifier, Standardization for preprocessing and Pearson correlation for feature selection.

But better results were obtained when relevance was predicted indirectly, based on an initial prediction of each of the six adopted criteria that followed a similar approach as the previous prediction of relevance. In a 10-fold cross validation, the best results of the second approach achieved an accuracy of 0.79 and a F_1 score of 0.82, using Random Forest classifiers for predicting each of the criteria and a kNN classifier for predicting relevance based on

those intermediate predictions. Besides reaching the best results, this approach confirmed that relevance can indeed be decomposed in the six aforementioned journalistic criteria.

The obtained results show a substantial improvement towards the only known work that tackled the same problem (Figueira et al, 2016), but where a much simpler feature set was exploited and a smaller dataset was used, to reach an accuracy of 0.59 and a F_1 of 0.68. Because of the different datasets used and, especially, the different classification goals and target classes, no deeper comparison can be made with related work, apart from a shallow analysis on the performance numbers reported. Having this in mind, our results are in line with, and in some cases higher than, other works that classify social web content according to a non-trivial aspect. This includes the detection spam tweets (Irani et al, 2010) ($F_1 = 0.79$); classification of appreciation ($F_1 = 0.78$), buzz ($F_1 = 0.81$), controversy ($F_1 = 0.80$), raising discussion ($F_1 = 0.68$) in messages (Guerini et al, 2011); predicting the number of retweets ($F_1 = 0.60$) (L. Hong and B. Davison, 2011); detecting satire ($F_1 = 0.89$) (Frain and Wubben, 2016); or predicting the popularity of web news articles (Fernandes et al, 2015) ($F_1 = 0.69$).

Several experiments were left to do and we believe that the presented results can still be further improved. For instance, our best results were obtained using Random Forest classifiers to predict each of the six criteria. However, this can be seen as a simplification, because it might not be the best choice for all criteria. This choice would have to be supported by a comparison of different classifiers for each of the six criteria. Among the linguistic features that were left unexplored, the extraction of facts in the form of *subject-predicate-object* triples could possibly improve the results; or dependency parsing, which would possibly improve the identification of *who* did *what*, *when*, *where* and *how*.

Furthermore, although the current goal was to exploit exclusively linguistic features, in the future, this approach will be compared with approaches that exploit other kinds of features, such as previous information about the author of the posts, their position in the network graph, as well as the number of likes and shares. In fact, resulting conclusions will support the integration of this work in a larger relevance mining platform that should soon be fully operating.

References

Fernandes K, Vinagre P, Cortez P (2015) A Proactive Intelligent Decision Support System for Predicting the Popularity of Online News. In: Progress in Artificial Intelligence, LNCS, vol 9273, Springer, pp 535–546

Figueira A, Sandim M, Fortuna P (2016) An approach to relevancy detection: Contributions to the automatic detection of relevance in social networks. In: New Advances in Information Systems and Technologies, Springer, pp 89–99

Frain A, Wubben S (2016) SatiricLR: a Language Resource of Satirical News Articles. In: Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016), European Language Resources Association (ELRA), Paris, France

Guerini M, Strapparava C, Özbal G (2011) Exploring text virality in social networks. In: International AAAI Conference on Web and Social Media

Irani D, Webb S, Pu C, Li K (2010) Study of Trend-stuffing on Twitter through Text Classification. In: Proceedings of 7th Collaboration, Electronic messaging, Anti-Abuse and Spam Conference (CEAS)

K Gimpel, N Schneider, B O'Connor, D Das, D Mills, J Eisenstein, M Heilman, D Yogatama, J Flanigan and N Smith (2011) Part-of-speech Tagging for Twitter: Annotation, Features, and Experiments. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2, Portland, Oregon, pp 42–47

K Lee, D Palsetia, R Narayanan, Md Ali, A Agrawal and A Choudhary (2011) Twitter Trending Topic Classification. In: Proceedings of the 2011 IEEE 11th International Conference on Data Mining Workshops, pp 251–258

L Hong and B Davison (2011) Predicting Popular Messages in Twitter. In: Proceedings of the 20th International Conference Companion on World Wide Web, Hyderabad, India, pp 57–58

Lops P, de Gemmis M, Semeraro G (2011) Content-based recommender systems: State of the art and trends. In: Recommender systems handbook, Springer US, pp 73–105

Nakov P, Rosenthal S, Kozareva Z, Stoyanov V, Ritter A, Wilson T (2013) SemEval-2013 Task 2: Sentiment Analysis in Twitter. In: Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), ACL Press, Atlanta, Georgia, USA, pp 312–320

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E (2011) Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12:2825–2830

Petrovic S, Osborne M, Lavrenko V (2011) RT to win! predicting message propagation in twitter. In: Proceedings of the Fifth International Conference on Weblogs and Social Media, Barcelona, Catalonia, Spain, July 17-21, 2011, The AAAI Press, URL `http://www.aaai.org/ocs/index.php/ICWSM/ICWSM11/paper/view/2754`

R Bandari, S Asur and B Huberman (2012) The Pulse of News in Social Media: Forecasting Popularity. In: Proceedings of the 6th International AAAI Conference on Web and Social Media, Dublin, Ireland, pp 26–33

Ritter A, Clark S, Etzioni O (2011) Named Entity Recognition in Tweets: An Experimental Study. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Edinburgh, Scotland, pp 1524–1534

Ritter A, Mausam, Etzioni O, Clark S (2012) Open domain event extraction from Twitter. In: Proceedings of 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, KDD'12, pp 1104–1112

Rose A (2015) Facebook is suffering an irrelevance crisis. http://www.marketingmagazine.co.uk/article/1371570/facebook-suffering-irrelevance-crisis, accessed: 06.11.2015

S Bird (2006) NLTK: The Natural Language Toolkit. In: Proceedings of the COLING/ACL on Interactive Presentation Sessions, Sydney, Australia, COLING-ACL '06, pp 69–72

Saracevic T (2015) Why is Relevance Still the Basic Notion in Information Science? (Despite Great Advances in Information Technology). In: Proceedings of the International Symposium on Information Science, Zadar, Croatia

Sriram B, Fuhry D, Demir E, Ferhatosmanoglu H, Demirbas M (2010) Short Text Classification in Twitter to Improve Information Filtering. In: Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, New York, NY, USA, SIGIR '10, pp 841–842, DOI 10.1145/1835449.1835643

Szabo G, Huberman BA (2010) Predicting the Popularity of Online Content. Communications of the ACM 53(8):80–88, DOI 10.1145/1787234.1787254

Tatar A, de Amorim MD, Fdida S, Antoniadis P (2014) A survey on Predicting the Popularity of Web Content. Journal of Internet Services and Applications 5(1):8:1–8:20, DOI 10.1186/s13174-014-0008-y

Wu M (2012) If 99.99% of Big Data is Irrelevant, Why Do We Need It? . https://community.lithium.com/t5/Science-of-Social-blog/If-99-99-of-Big-Data-is-Irrelevant-Why-Do-We-Need-It/ba-p/39310, accessed: 06.11.2015

Yu B, Chen M, Kwok L (2011) Toward Predicting Popularity of Social Marketing Messages. In: Social Computing, Behavioral-Cultural Modeling and Prediction, Lecture Notes in Computer Science, vol 6589, Springer Berlin Heidelberg, pp 317–324

Zeng YC, Wu SH (2013) Modeling the Helpful Opinion Mining of Online Consumer Reviews as a Classification Problem. In: Proceedings of the IJCNLP 2013 Workshop on Natural Language Processing for Social Media (SocialNLP), Asian Federation of Natural Language Processing, Nagoya, Japan, pp 29–35