Automatic Detection of Political Opinions in Tweets

Diana Maynard and Adam Funk

Department of Computer Science, University of Sheffield
Regent Court, 211 Portobello Street, Sheffield, UK
`diana@dcs.shef.ac.uk`

Abstract. In this paper, we discuss a variety of issues related to opinion mining from microposts, and the challenges they impose on an NLP system, along with an example application we have developed to determine political leanings from a set of pre-election tweets. While there are a number of sentiment analysis tools available which summarise positive, negative and neutral tweets about a given keyword or topic, these tools generally produce poor results, and operate in a fairly simplistic way, using only the presence of certain positive and negative adjectives as indicators, or simple learning techniques which do not work well on short microposts. On the other hand, intelligent tools which work well on movie and customer reviews cannot be used on microposts due to their brevity and lack of context. Our methods make use of a variety of sophisticated NLP techniques in order to extract more meaningful and higher quality opinions, and incorporate extra-linguistic contextual information.

Keywords: NLP, opinion mining, social media analysis.

1 Introduction

Social media provides a wealth of information about a user's behaviour and interests, from the explicit "John's interests are tennis, swimming and classical music", to the implicit "people who like skydiving tend to be big risk-takers", to the associative "people who buy Nike products also tend to buy Apple products". While information about individuals is not always useful on its own, finding defined clusters of interests and opinions can be interesting. For example, if many people talk on social media sites about fears in airline security, life insurance companies might consider opportunities to sell a new service. This kind of predictive analysis is all about understanding one's potential audience at a much deeper level, which can lead to improved advertising techniques, such as personalised advertisements to different groups.

It is in the interests of large public knowledge institutions to be able to collect and retrieve all the information related to certain events and their development over time. In this new information age, where thoughts and opinions are shared through social networks, it is vital that, in order to make best use of this information, we can distinguish what is important, and be able to preserve it, in order

to provide better understanding and a better snapshot of particular situations. Online social networks can also trigger a chain of reactions to such situations and events which ultimately lead to administrative, political and societal changes.

In this paper, we discuss a variety of issues related to opinion mining from microposts, and the challenges they impose on a Natural Language Processing (NLP) system, along with an example application we have developed to divulge political leanings from a set of pre-election tweets. While knowing that Bob Smith is a Labour supporter is not particularly interesting on its own, when this information is combined with other metadata, and information about various groups of people is combined and analysed, we can begin to get some very useful insights about political leanings and on factors that impact this, such as debates aired on television or political incidents that occur.

We first give in Section 2 some examples of previous work on opinion mining and sentiment analysis, and show why these techniques are either not suitable for microposts, or do not work particularly well when adapted to other domains or when generalised. We then describe the opinion mining process in general (Section 3), the corpus of political tweets we have developed (Section 4), and the application to analyse opinions (Section 5). Finally, we give details of a first evaluation of the application and some discussion about future directions (Sections 6 and 8).

2   Related Work

Sentiment detection has been applied to a variety of different media, typically to reviews of products or services, though it is not limited to these. Boiy and Moens [1], for example, see sentiment detection as a classification problem and apply different feature selections to multilingual collections of digital content including blog entries, reviews and forum postings. Conclusive measures of bias in such content have been elusive, but progress towards obtaining reliable measures of sentiment in text has been made – mapping onto a linear scale related to positive versus negative, emotional versus neutral language, etc.

Sentiment detection techniques can be roughly divided into lexicon-based methods [2] and machine-learning methods [1]. Lexicon-based methods rely on a sentiment lexicon, a collection of known and pre-compiled sentiment terms. A document's polarity is the ratio of positive to negative terms. Machine learning approaches make use of syntactic and/or linguistic features, including sentiment lexicons. Hybrid approaches are very common, and sentiment lexicons play a key role in the majority of methods. However, such approaches are often in-flexible regarding the ambiguity of sentiment terms. The context in which a term is used can change its meaning, which is particularly true for adjectives in sentiment lexicons [3]. Several evaluations have shown that sentiment detection methods should not neglect contextual information [4,5], and have identified context words with a high impact on the polarity of ambiguous terms [6]. Besides the ambiguity of human language, another bottleneck for sentiment detection methods is the time-consuming creation of sentiment dictionaries. One solution

to this is a crowdsourcing technique to create such dictionaries with minimal effort, such as the Sentiment Quiz Facebook application[1].

However, sentiment dictionaries alone are not enough, and there are major problems in applying such techniques to microposts such as tweets, which typically do not contain much contextual information and which assume much implicit knowledge. They are also less grammatical than longer posts and make frequent use of emoticons and hashtags, which can form an important part of the meaning. This means that typical NLP solutions such as full - or even shallow - parsing are unlikely to work well, and new solutions need to be incorporated for handling extra-linguistic information. Typically, they also contain extensive use of irony and sarcasm, which are also difficult for a machine to detect.

There exists a plethora of tools for performing sentiment analysis of tweets, though most work best on mentions of product brands, where people are clearly expressing opinions about the product. Generally, the user enters a search term and gets back all the positive and negative (and sometimes neutral) tweets that contain the term, along with some graphics such as pie charts or graphs. Typical basic tools are Twitter Sentiment[2], Twends[3] and Twitrratr[4]. Slightly more sophisticated tools such as SocialMention[5] allow search in a variety of social networks and produce other statistics such as percentages of Strength, Passion and Reach, while others allow the user to correct erroneous analyses. While these tools are simple to use and often provide an attractive display, their analysis is very rudimentary, performance is low, and they do not identify the opinion holder or the topic of the opinion, assuming (often wrongly) that the opinion is related to the keyword.

3   Opinion Mining Process

We have developed an initial application for opinion mining using GATE [7], a freely available toolkit for language processing. The first stage in the system is to perform a basic sentiment analysis, i.e., to associate a positive, negative or neutral sentiment with each relevant tweet. This is supplemented by creating triples of the form <*Person, Opinion, Political Party*>, e.g., <*Bob Smith, pro, Labour*> to represent the fact that Bob Smith is a Labour supporter. Given the nature of tweets, we have found that it is fairly rare to see more than one sentiment about the same thing expressed in a single tweet: if, however, two opposing opinions about a political party are mentioned, then we simply posit a neutral opinion at this stage.

Once the triples have been extracted, we can then collect all mentions that refer to the same person, and collate the information. For example, John may be equally drawn towards more than one party, not just Labour, but hates

---

[1] http://apps.facebook.com/sentiment-quiz
[2] http://twittersentiment.appspot.com/
[3] http://twendz.waggeneredstrom.com/
[4] http://twitrratr.com/
[5] http://socialmention.com/

the Conservatives. His opinion may also change over time, especially during the pre-election phase, or since the recent elections. We thus go beyond typical sentiment analysis techniques which only look at a static opinion at a fixed point in time. This is important because it enables us to make much more interesting observations about political opinions and how they are affected by various events.

4   The Pre-election Twitter Corpus

For the development of our application, we used a corpus of political tweets collected over the UK pre-election period in 2010[6]. The Twitter Streaming API[7] was used to collect tweets from this period according to a variety of relevant criteria (use of hash tags such as #election2010, #bbcqt (BBC Question Time), #Labour etc., specific mention of various political parties or words such as "election", and so on). The tweets were collected in JSON format and then converted to xml using the JSON-Lib library[8]. The corpus contains about 5 million tweets; however it contains many duplicates. De-duplication, which formed a part of the conversion process, reduced the corpus size by about 20% to around 4 million tweets.

The corpus contains not only the tweets themselves, but also a large amount of metadata associated with each tweet, such as its date and time, the number of followers of the person tweeting, the location and other information about the person tweeting, and so on. This information is useful for disambiguation and for collating the information later. Figure 1 depicts a tweet loaded in GATE, with the text and some of the metadata (location, author, and author profile) highlighted. We should note that the method for collecting tweets is not perfect, as we find some tweets which are nothing to do with the election, due to ambiguous words (in particular, "Labour" which could also be a common noun, and "Tory" which could also be a person's name). For future work, we plan a more sophisticated method for collecting and pruning relevant tweets; nevertheless, this quick and dirty method enabled us to get the initial experiments underway quickly.

5   Application

The application consists of a number of processing modules combined to form an application pipeline. First, we use a number of linguistic pre-processing components such as tokenisation, part-of-speech tagging, morphological analysis, sentence splitting, and so on. Full parsing is not used because of the nature of the tweets: from past experience, we know it is very unlikely that the quality would be sufficiently high. Second, we apply ANNIE [8], the default named entity recognition system available as part of GATE, in order to recognise named entities in the text (Person, Organisation, Location, Date, Time, Money, Percent).

---

[6] We are very grateful to Matthew Rowe for allowing us to use this corpus.
[7] http://dev.twitter.com/pages/streaming_api
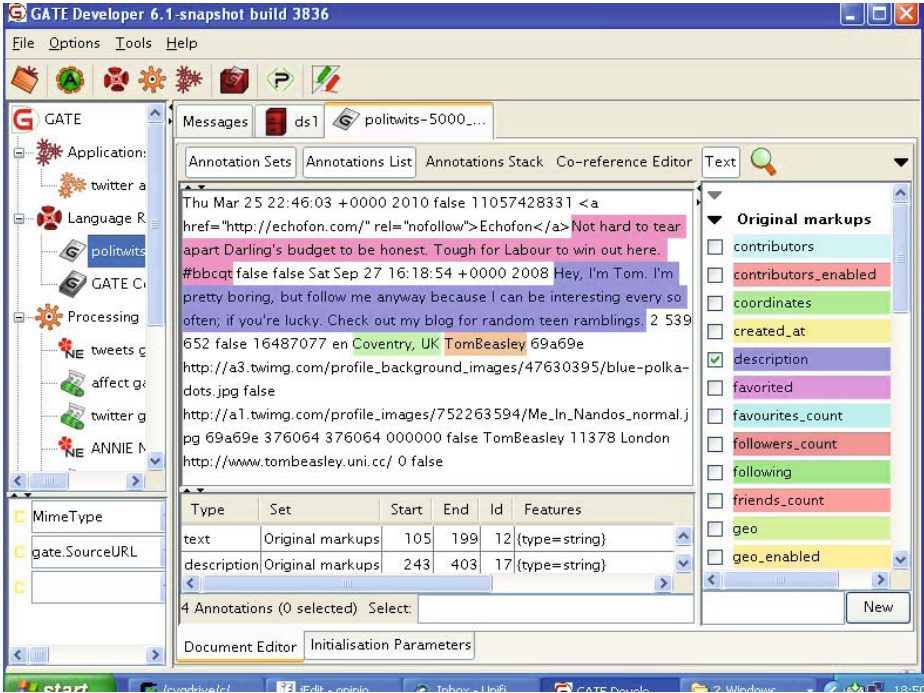[8] http://json-lib.sourceforge.net/

Fig. 1. Screenshot of a tweet in GATE, with relevant metadata

The named entities are then used in the next stages: first for the identification of opinion holders and targets (i.e., people, political parties, etc.), and second, as contextual information for aiding opinion mining.

The main body of the opinion mining application involves a set of JAPE grammars which create annotations on segments of text. JAPE is a Java-based pattern matching language used in GATE [9]. The grammar rules create a number of temporary annotations which are later combined with existing annotations and converted into final annotations. In addition to the grammars, we use a set of gazetteer lists containing useful clues and context words: for example, we have developed a gazetteer of affect/emotion words from WordNet[10]. These have a feature denoting their part of speech, and information about the original WordNet synset to which they belong. The lists have been modified and extended manually to improve their quality: some words and lists have been deleted (since we considered them irrelevant for our purpose) while others have been added.

As mentioned above, the application aims to find for each relevant tweet, triples denoting three kinds of entity: Person, Opinion and Political Party. The application creates a number of different annotations on the text which are then combined to form these triples.

The detection of the actual opinion (sentiment) is performed via a number of different phases: detecting positive, negative and neutral words (Affect annotations), identifying factual or opinionated versus questions or doubtful statements, identifying negatives, and detecting extra-linguistic clues such as smileys.

Because we only want to process the actual text of the tweet, and not the metadata, we use a special processing resource (the *Segment Processing PR*) to run our application over just the text covered by the XML "text" tag in the tweet. We also use this to access various aspects of the metadata from the tweet, such as the author information, as explained below.

5.1   Affect Annotations

Affect annotations denote the sentiment expressed in the tweet, which could be positive, negative or neutral towards a particular party. These are created primarily by the gazetteer (sentiment dictionary), but the sentiment denoted can then be modified by various contextual factors. First, the gazetteer is used to find mentions of positive and negative words, such as "beneficial" and "awful" respectively. A check is performed to ensure that the part of speech of the gazetteer entry matched and the word in the text are the same, otherwise no match is produced. This ensures disambiguation of the various categories. For example, note the difference between the following pairs of phrases: "Just watched video about awful days of Tory rule" vs "Ah good, the entertainment is here." In the first phrase, "awful" is an adjective and refers to the "days of Tory rule": this would be appropriate as a match for a negative word. In the second phrase, "good" is an adverb and should not be used as a match for a positive sentiment about the entertainment (it does not actually denote that the entertainment itself is good, only that the author is looking forward to the entertainment). Similarly, note the difference between the preposition and verb "like" in the following pair of phrases, which again express very different sentiments about the person in question: "People like her should be shot." vs "People like her."

5.2   Other Clues

Hashtags can also be a source of information about the opinion of the author. Some are fairly explicit, for example #VoteSNP, #Labourfail, while others are more subtle, e.g., #torytombstone, #VoteFodderForTheTories. Currently, we list a number of frequently occurring hashtags of this sort in a gazetteer list, but future work will involve deconstructing some of these hashtags in order to deduce their meaning on the fly (since they are not correctly tokenised, they will not be recognised by our regular gazetteers and grammars). Some hashtags are easier to decipher the meaning of than others: for example, #torytombstone requires some implicit knowledge about the use of the word "tombstone" being used in a derogatory way.

5.3   Opinionated Statements

This phase checks the tweets to see if they are opinionated, or whether they contain questions or doubtful statements. For example, it is hard to tell from the

question: "Wont Unite's victory be beneficial to Labour?" whether the author is a supporter of Labour or not, so we posit simply a neutral opinion here. Initially, we match any statement containing an Affect annotation as being opinionated, unless it contains a question, but this could be extended to deal with other cases. We annotate any tweet that contains a question mark (or potentially other distinguishing question-related features) as a Question, and retain it for possible later use, but do not annotate it as an Opinion at this point.

5.4   Negatives

This phase checks the tweet to see if it contains some negative word (as found in a gazetteer list), such as "not", "couldn't", "never" and so on. In most cases, it will reverse the opinion already found: the existing feature value on the Sentiment annotation is changed from "pro" to "anti" or vice versa. More complex negatives include checking for words such as "against", "stop" and so on as part of a noun phrase involving a political party, or as part of a verb phrase followed by a mention of a political party.

5.5   Political Party

Finding the name of the Political Party in the tweet is generally straightforward as there are only a limited number of ways in which they are usually referred to. As mentioned above, however, there is some ambiguity possible. We therefore use other clues in the tweet, where possible, to help resolve these. For example, if "Tory" is part of a person's name (identified by ANNIE), we discard it as a possible political party.

5.6   Identifying the Opinion Holder

Where a Person is identified in the tweet as the holder of the opinion (via another set of grammar rules), we create a Person annotation. If the opinion holder in the pattern matched is a Person or Organization, we create a Person annotation with the text string as the value of an opinion_holder feature on the annotation. If the opinion holder in the pattern matched is a pronoun, we first find the value of the string of the matching proper noun antecedent and use this as the value of the opinion_holder feature. Currently, we only match opinion holders within the same sentence.

However, there may not always be an explicit opinion holder. In many cases, the author of the tweet is the opinion holder, e.g., "I'm also going to vote Tory. Hello new world." Here we can co-refer "I" with the person tweeting. In other cases, there is no opinion holder explicitly mentioned, e.g., "Vote for Labour. Harry Potter would." In this case, we can assume that the opinion is also held by the author. In both cases, therefore, we use "author" as the value of opinion_holder, and get the details of the tweet author from the xml metadata.

5.7   Creating Triples

As described above, we first create temporary annotations for Person, Organization, Vote, Party, Negatives etc. based on gazetteer lookup, named entities and so on. We then use a set of rules to combine these into triples, for example:

<Person, Vote, Party>
"Tory Phip admits he voted LibDem" → *<Phip, pro, LibDem>*

<Person, Party, Affect>
"When they get a Tory government they'll be sorry." → *<author, anti, Tory>*

Finally, we create an annotation "Sentiment" which has the following features:

– kind = pro_Labour, anti_LibDem, etc.
– opinion_holder = Bob Smith, author, etc.

Currently, we restrict ourselves to rules which are very likely to be successful, thus achieving high Precision at the expense of Recall. These rules should be eventually expanded in order to get more hits, although Precision may suffer as a result.

6   Evaluation and Discussion

We evaluated the first stage of this work, i.e., the basic opinion finding application, on a sample set of 1000 tweets from the large political corpus (selected randomly by a Python script). We then ran the application over this test set and compared the results. Table 1 gives some examples of the different opinions recognised by the system: it shows the tweet (or the relevant portion of the tweet), the opinion generated by the system (labelled "System") and the opinion generated by the manual annotator (labelled "Key").

Out of 1000 tweets, the system identified 143 as being opinionated (about a political party), i.e., it created a Sentiment annotation for that tweet. We analysed these tweets manually and classified them into the following categories: ProCon, AntiCon, ProLab, AntiLab, ProLib, AntiLib, Unknown and Irrelevant. The first 6 of these categories match the system annotations. Unknown is marked when either a political opinion is not expressed or where it is unclear what the political opinion expressed is, e.g., *"Labour got less this time than John Major did in 1997."* Irrelevant is marked when the tweet is not relevant to politics or the election, e.g., *"i am soooooooooo bored, want to go into labour just for something to do for a couple of hours :)"*. The distinction between Irrelevant and Unknown is only important in that it tells us which tweets should ideally be excluded from the corpus before analysis: we want to include the Unknown ones in the corpus (even though the system should not annotate them), in order to ensure that the system does not annotate false positives as containing a political sentiment, but not the Irrelevant ones. While only 2 documents out of the 143 were classed as Irrelevant, 29 were classed as Unknown (roughly 20%). This means that roughly

Table 1. Examples of tweets and the opinions generated

Tweet	System	Key
I just constantly said "Vote Labour" in a tourettes kinda way	pro-Lab	pro-Lab
Daily Mail reveals PM's wife has ugly feet http://bit.ly/b6ZNlK ¡–Eww! Another reason not to vote Labour.	pro-Lab	pro-Lab
Still, can't bring myself to vote tactically for Labour	anti-Lab	anti-Lab
@WilliamJHague If you fancy Interest Rates at 15.8% Vote Tory .... they will throw you out of your house...back to the 80's	pro-Con	anti-Con
Vote Tory to stop them bleating! You know it's worth it.	pro-Con	pro-Con
George Osborne. Reason number 437 not to vote Tory.	anti-Con	anti-Con
Vote Tory or Labour, get Lib Dems. Might as well vote LibDem and have done with it	pro-Lib	pro-Lib
@Simon_Rayner sorry but laughing so much it hurts. Who in their right mind will vote for libdem savage cuts?	anti-Lib	anti-Lib

Table 2. Confusion matrix for evaluation corpus

Key/System	ProCon	AntiCon	ProLab	AntiLab	ProLib	AntiLib	Total
ProCon	**5**	0	0	0	0	0	5
AntiCon	10	**5**	0	2	0	0	17
ProLab	0	0	**69**	2	0	0	70
AntiLab	0	0	4	**4**	0	0	8
ProLib	3	0	1	0	**6**	0	10
AntiLib	0	0	0	0	0	**1**	1
Unknown	10	1	11	5	2	0	29
Irrelevant	0	1	0	1	0	0	2
Total	28	7	85	14	8	1	143

80% of the documents that the system classified with a Sentiment, were in fact opinionated, though not all of them had the correct opinion.

Table 2 shows a confusion matrix for the different sentiments recognised by the system, compared with those in the Key (the manually generated sentiments). This table only depicts the results for those tweets where the system recognised a Sentiment as present: it does not show the Missing annotations (where the system failed to recognise a valid Sentiment). The figures in bold indicate a correct match between System and Key. Overall, the system achieved a Precision of 62.2%, which is promising for work at this early stage.

Unfortunately, it was not feasible in this preliminary evaluation to manually annotate 1000 tweets, so we cannot calculate system Recall easily. However, we can extrapolate some hypothesised Recall based on a smaller sample. We took 150 of the tweets which were not classified as opinionated by the system, and annotated them manually. Of these, 127 (85%) were correct, i.e., were classified as Unknown or Irrelevant by the human annotator. Assuming that this sample is representative, we can predict the Recall. For the task of finding whether a political sentiment exists or not (regardless of its orientation), we get 78%

Precision and predict 47% Recall. Where a document was found to contain a political sentiment, the polarity of this sentiment was correct in 79% of cases. Overall, for the task of both correctly identifying that a document contained a political sentiment, and correctly identifying its polarity, we get 62% Precision and predict 37% Recall.

While the Recall of our system is clearly less than ideal, this is unsurprising at this stage because it has been developed with Precision rather than Recall in mind, i.e., only to produce a result if it is reasonably certain. As we have discussed earlier, there is plenty of scope for improvements to the NLP, in order to improve the Recall of the system. The Precision could also be tightened up further by improving the negation aspect of the rules (most of the errors are related either to not correctly identiyying a negative, or by missing out on language nuances such as sarcasm, which are hard for an automated system to deal with). Further evaluation will focus on a larger number of tweets.

It is important also to recognise in the context of evaluation, that performing NLP tasks on social media is in general a harder task than on news texts, for example, because of the style and lack of correct punctuation, grammar etc. Shorter posts such as tweets suffer even more in this respect, and therefore performance of NLP is likely to be lower than for other kinds of text. Also, tweets in particular assume a high level of contextual and world knowledge by the reader, and this information can be very difficult to acquire automatically. For example, one tweet in our dataset likened a politician to Voldemort, a fictional character from the Harry Potter series of books. This kind of world knowledge is unlikely to be readily available in a knowledge base for such an application, and we may have to just accept that this kind of comment cannot be readily understood by auomatic means (unless we have sufficient examples of it occurring).

7   Prospects for Machine Learning

In previous work [11,12] we have obtained good results using SVM-based machine learning (ML) from linguistic features for opinion classification. We plan to experiment with similar data-driven techniques on tweets, although we would probably use the Perceptron algorithm instead, since it is faster and (in our experience) about as accurate for NLP. Our previous experiments were carried out on longer, somewhat more consistently edited texts (film, product and business reviews)—quite unlike the highly abbreviated and inconsistent styles found in tweets—but we obtained good results with unigrams of simple linguistic features, such as tokens and their lemmas, as well as with features derived from SentiWordNet values.

To carry out such experiments successfully on tweets, however, we would need a larger manually annotated corpus—probably 500 tweets annotated with the same 8-way classification used above. This figure is based on our normal use of 5- or 10-fold cross-validation (to stretch the training and test data) for ML as well as on the difficulty of classifying very short texts. We would not necessarily expect the SentiWordNet features to help much with tweets, which are more polysemous and often sarcastic, but it would be interesting to compare them.

8   Conclusions

Typically, opinion mining looks at social media content to analyse people's explicit opinions about an organisation, product or service. However, this backwards-looking approach often aims primarily at dealing with problems, e.g., unflattering comments, while a forwards-looking approach aims at looking ahead to understanding potential new needs from consumers. This is achieved by trying to understand people's needs and interests in a more general way, e.g. drawing conclusions from their opinions about other products, services and interests. It is not sufficient, therefore, to look at specific comments in isolation: non-specific sentiment is also an important part of the overall picture.

One of the difficulties of drawing conclusions from traditional opinion mining techniques is the sparse data issue. Opinions about products and services tend to be based on one very specific thing, such as a particular model of camera or brand of washing powder, but do not necessarily hold for every other model of that brand of camera, or for every other product sold by the company, so a set of very isolated viewpoints is typically identified. The same applies, in some sense, to political viewpoints: a person may not like a particular politician even if they support the party represented by that person, overall. Furthermore, political opinions are often more suject to variations along a timeline than products and brands. A person who prefers Coke to Pepsi is unlikely to change their point of view suddenly one day, but there are many people whose political leanings change frequently, depending on the particular government, the politicians involved and events which may occur. In order to overcome such issues, we need to be able to figure out which statements can be generalised to other models/products/issues, and which are specific. Another solution is to leverage sentiment analysis from more generic expressions of motivation, behaviour, emotions and so on, e.g., what type of person buys what kind of camera, what kind of person is a Labour supporter, and so on. This requires combining the kind of approach to opinion mining described here with additional information about people's likes, dislikes, interests, social groups and so on. Such techniques will form part of our future work.

As discussed earlier, there are many improvements which can be made to the opinion mining application in terms of using further linguistic and contextual clues: the development of the application described here is a first stage towards a more complete system, and also contextualises the work within a wider framework of social media monitoring which can lead to interesting new perspectives when combined with relevant research in related areas such as trust, archiving and digital libraries. In particular, the exploitation of Web 2.0 and the wisdom of crowds can make web archiving a more selective and meaning-based process. Analysis of social media can help archivists select material for inclusion, providing content appraisal via the social web, while social media mining itself can enrich archives, moving towards structured preservation around semantic categories.

References

1. Boiy, E., Moens, M.F.: A machine learning approach to sentiment analysis in multilingual web texts. Information Retrieval 12(5), 526–558 (2009)
2. Scharl, A., Weichselbraun, A.: An automated approach to investigating the online media coverage of US presidential elections. Journal of Information Technology and Politics 5(1), 121–132 (2008)
3. Mullaly, A., Gagné, C., Spalding, T., Marchak, K.: Examining ambiguous adjectives in adjective-noun phrases: Evidence for representation as a shared core-meaning. The Mental Lexicon 5(1), 87–114 (2010)
4. Weichselbraun, A., Gindl, S., Scharl, A.: A context-dependent supervised learning approach to sentiment detection in large textual databases. Journal of Information and Data Management 1(3), 329–342 (2010)
5. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing contextual polarity: An exploration of features for phrase-level sentiment analysis. Computational Linguistics 35(3), 399–433 (2009)
6. Gindl, S., Weichselbraun, A., Scharl, A.: Cross-domain contextualisation of sentiment lexicons. In: Proceedings of 19th European Conference on Artificial Intelligence (ECAI 2010), pp. 771–776 (2010)
7. Cunningham, H., Maynard, D., Bontcheva, K., Tablan, V.: GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications. In: Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics, ACL 2002 (2002)
8. Maynard, D., Tablan, V., Cunningham, H., Ursu, C., Saggion, H., Bontcheva, K., Wilks, Y.: Architectural Elements of Language Engineering Robustness. Journal of Natural Language Engineering – Special Issue on Robust Methods in Analysis of Natural Language Data 8(2/3), 257–274 (2002)
9. Cunningham, H., Maynard, D., Tablan, V.: JAPE: a Java Annotation Patterns Engine 2 edn. Research Memorandum CS–00–10, Department of Computer Science, University of Sheffield (2000)
10. Miller, G.A., Beckwith, R., Felbaum, C., Gross, D., Miller, C., Miller, G.A., Beckwith, R., Felbaum, C., Gross, D., Miller, C., Minsky, M.: Five papers on WordNetk-lines: A theory of memory (1980)
11. Funk, A., Li, Y., Saggion, H., Bontcheva, K., Leibold, C.: Opinion analysis for business intelligence applications. In: First international workshop on Ontology-Supported Business Intelligence (at ISWC). ACM, Karlsruhe (2008)
12. Saggion, H., Funk, A.: Extracting opinions and facts for business intelligence. RNTI Journal E(17), 119–146 (2009)