Heart and Soul: Sentiment Strength Detection in the Social Web with SentiStrength[1]

Mike Thelwall
Statistical Cybermetrics Research Group, School of Technology, University of Wolverhampton, Wulfruna Street, Wolverhampton WV1 1SB, UK.

Emotions are important in communication to effectively convey messages and to understand reactions to messages. Large scale studies of communication need methods to detect sentiment in order to investigate or model the processes involved. This chapter describes the sentiment strength detection program SentiStrength that was developed during the CyberEmotions project to detect the strength of sentiments expressed in social web texts. SentiStrength uses a lexical approach that exploits a list of sentiment-related terms and has rules to deal with standard linguistic and social web methods to express sentiment, such as emoticons, exaggerated punctuation and deliberate misspellings. This chapter also describes how SentiStrength can be refined for particular topics and contexts and how variants are created for different languages. The chapter also briefly describes some studies that have applied SentiStrength to analyse trends in Twitter and You Tube comments.

Introduction

Emotions and sentiments are critical to many human activities, including communication. People not only engage in social communication because they enjoy it or because it helps to fulfil emotional needs, but they also use sentiment to help convey meaning and react to sentiments expressed towards them or others. Hence, those seeking to model or understand communication patterns on a large scale need to account for the emotions of the participants or at least the sentiments expressed in their messages.

The importance of emotions applies not only to real time face-to-face communication. It is well documented that people can feel and express emotions through computer mediated communication (CMC) even if it is asynchronous and text-based (Walther & Parks, 2002). For example emoticons arose as a partial solution to the lack of body language and intonation to express emotion in informal types of text-based CMC (Derks, Bos, & von Grumbkow, 2008). Hence, to effectively analyse any area of the social web, emotion should be taken into account for all except the simplest models. If using real data for such analyses, it is necessary to have an automatic method to extract sentiment from text and this is the sentiment analysis task.

Sentiment analysis software reads text and uses an algorithm to produce an estimate of its sentiment content. This estimate can be in several different forms: binary – either positive/negative or objective/subjective; trinary – positive/neutral negative; scale – e.g., 5 (strongly negative) to 5 (strongly positive); dual scale – e.g., 1 (no positivity) – 5 (strong positivity) and -1 (no negativity) - -5 (strong negativity); and multiple – e.g., happiness (0-100), sadness (0-100), fear (0-100). Sentiment analysis algorithms tend to use either a machine learning or a lexical approach. A machine learning approach may start by converting each text into a list of words, consecutive word pairs and consecutive word

[1] To appear in Holyst, J. (Ed). Cyberemotions.

triples (i.e., 1-3grams) and then, based upon a human coded set of texts, 'learn' which of these features tend to associate with sentiment scores, using this information to classify new cases. In contrast, a lexical approach may start with some language information, such as a list of sentiment words and their polarities, and use this information together with grammatical structure knowledge, such as the role of negation, to estimate the sentiment of texts. To illustrate the difference between the two, a machine learning approach may classify "I am not happy" as negative because the bigram "not happy" occurs almost always in texts in the training set coded as negative by humans whereas the lexical approach may choose negative because "happy" is a known positive word and "not" is a known negating word that occurs immediately before it. The two approaches seem to have similar levels of accuracy (however measured) depending upon the types of texts classified and the amount of human classified training data available. Nevertheless, lexical sentiment analysis seems to be superior from a pragmatic perspective for many social research applications because it is less likely to pick up indirect indicators of sentiment that will generate spurious sentiment patterns. For instance a machine learning approach might extract unpopular politicians' names as negative features because they tend to occur in negative texts but this would result in even objective or neutral texts about them being classified as negative, undermining any derived analysis of sentiment in political communication.

This chapter describes SentiStrength, a free sentiment analysis program that uses a lexical approach to classify social web texts. It uses the dual positive – negative scales because psychological research reports that humans can experience positive and negative emotions simultaneously and to some extent independently (Norman et al., 2011). It also uses the lexical approach for the reasons given above and harnesses CMC conventions for expressing sentiment to capture non-standard expressive text. As the results below show, it works well without any training data on a wide range of social web texts and approaches human-level accuracy in most tested cases. The exceptions where it performs less well are sets of texts with widespread irony or sarcasm, such as informal political discussions and narrowly-focused topics with frequently used sentiment terms that are either rare in other topics or tend to have a different meaning.

This chapter explains in detail how SentiStrength works, reports evaluations of it on Twitter, YouTube and other data sets, describes how to customise it for specific topics or topics with a negative mood, introduces an extension to enhance its accuracy within sets of related texts, explains how to customise it for languages other than English and reports analyses of Twitter and YouTube with it.

Using SentiStrength

SentiStrength is available in two versions, Java and Windows. The Windows version can be downloaded free from the website http://sentistrength.wlv.ac.uk/ and the Java version is available for purchase from the author for commercial users or free for researchers and educational users. There is also an interface on the SentiStrength web site to try out SentiStrength live online. The web site includes the main English version as well as several other language variants. SentiStrength's commercial users include Yahoo! (Kucuktunc, Cambazoglu, Weber, & Ferhatosmanoglu, 2012; Weber, Ukkonen, & Gionis, 2012) and a range of online information management companies around the world. It was also used to power a light display on the EDF Energy London Eye during the London 2012 Olympic Games by continually monitoring the average sentiment of Olympic-related tweets. The Java version can process 16,000 tweets per second on a standard PC and can be configured to output dual, scale, binary and trinary results (as described above).

The SentiStrength resources, such as the sentiment lexicon and emoticon list, are stored as separate text files and SentiStrength must be pointed to the location when started. It can process text in various ways (depending upon the version), including: single texts submitted via the command line, batches of texts in a single or multiple plain text files with each line of a file classified separately; listening on an internet port; and reading stdin.

The core SentiStrength algorithm

The heart of SentiStrength is a lexicon of 2310 sentiment words and word stems obtained from the Linguistic Inquiry and Word Count (LIWC) program (Pennebaker, Mehl, & Niederhoffer, 2003), the General Inquirer list of sentiment terms (Stone, Dunphy, Smith, & Ogilvie, 1966) and ad-hoc additions made during testing, particularly for new CMC words. The (Kleene star) stemming used is simple and indicated in the lexicon with a wildcard at the end of a word. For instance *amaz** matches all words starting with amaz, such as amazed and amazing. For each text, SentiStrength outputs a positive sentiment score from 1 to 5 and a negative score from -1 to -5. Matching this, each word or stem in the dictionary is given a positive or negative score within one of these two ranges. These scores were initially human assigned based upon a development corpus of 2,600 comments from the social network site MySpace, and subsequently updated through additional testing. The weights for the terms in the sentiment lexicon have been tested against several data sets and can be fine-tuned by SentiStrength using a machine learning approach, as discussed below. The reason for primarily relying upon human input for the sentiment weights is that many of the terms occur rarely in texts and so a machine learning approach to assign weights well would need a huge number of classified texts to give sufficient coverage for the lexicon. This is a long tail effect because even though many individual terms in the lexicon are rare, collectively the rare terms occur often enough to affect the performance of SentiStrength.

The lexicon is used in a simple way. When SentiStrength reads a text, it splits it into words and separates out emoticons and punctuation. Each word is then checked against the lexicon for matching any of the sentiment terms. If a match is found then the associated sentiment score is retained. The overall score for a sentence is the highest positive and negative score for its constituent words and for multiple sentences and the maximum scores of the individual sentences is taken. For example, the text "Mike is horrible and nasty but I am lovely. I am fantastic." would be classified as follows, "Mike is horrible[-4] and nasty[-3] but I am lovely[2]. <sentence score: 2,-4> I am fantastic[3]. <sentence: 3,-1>" with numbers in square brackets indicating sentiment strength of the preceding word, and angle brackets indicate sentence scores. The overall classification for this text is the maximum positive and negative strength of each sentiment, which is 3 and -4.

The above scores are kept unless they are modified by any of SentiStrength's additional rules. An odd feature of the lexicon is that it contains some non-sentiment terms with a score of 1 (no positivity) or -1 (no negativity). There are two reasons for these term, some are included to match non-sentiment variants of sentiment stems. For instance *amaz** is a positive stem but *amazon* is added as a neutral term to catch this non-sentiment term that would otherwise match *amaz**. This works because SentiStrength returns the longest matching term in cases of multiple matches. Some neutral terms are also included as reminders that they have been assessed for inclusion in the lexicon and rejected.

In addition to the lexicon, SentiStrength includes a list of emoticons together with human-assigned sentiment scores. Emoticons are somewhat tricky to automatically extract because although they are typically constructed from lists of punctuation characters and surrounded by spaces, some contain numbers or letters and they may be followed by punctuation that is not part of the emoticon. Hence emoticon extraction is imperfect. SentiStrength also has a list of idioms with sentiment strength weights. These are all multiple word phrases with a meaning that is different from their component word. These idiom scores override the lexicon scores. For example, the stock phrase “shock horror” has an idiom score of -2 for mildly negative and overrides the strong negative scores for shock (-3) and horror (-4).

A weakness of SentiStrength is that it does not attempt to use grammatical parsing (e.g., part of speech tagging) to disambiguate between different word senses. This is because it is designed to process very informal text from the social web and so, unlike typical linguistic parsers, does not rely upon standard grammar for optimal performance. Some grammatical information is used by SentiStrength, however, as the rules below show, and the idiom table can also be used for a brute force approach. To illustrate this, the word “like” can express positive sentiment (e.g., “I like you”) or can be used as a comparator (e.g., “I look like an idiot”). SentiStrength gives a neutral score to *like* but has phrases containing *like* in its idiom list with a positive score to override the neutral score for *like* when it is used in a common positive way (e.g. “he likes”, “I like”, “we like”, “she likes”).

Additional Sentiment Rules

In addition to the sentiment term strength lexicon, the idiom list and the emoticon list, SentiStrength incorporates a number of rules to cope with special cases. These were mainly derived from testing on the 2600 My Space comments development data set by examining cases of wrong scores given by early visions of SentiStrength and formulating general rules to cope with them. The following rules are incorporated into SentiStrength (M. Thelwall, Buckley, & Paltoglou, 2012).

- An **idiom list** is used to identify the sentiment of a few common phrases. This overrides individual sentiment word strengths. The idiom list is extended with phrases indicating word senses for common sentiment words, as described above.
- The word “**miss**” is a special case with a positive strength of 2 and a negative strength of -2. It is frequently used to express sadness and love simultaneously, as in the common phrase, "I miss you".
- A **spelling correction algorithm** deletes repeated letters in a word when the letters are more frequently repeated than normal for English or, if a word is not found in an English dictionary, when deleting repeated letters creates a dictionary word (e.g., hellp -> help).
- **At least two repeated letters** added to words give a strength boost sentiment words by 1. For instance haaaappy is more positive than happy. Neutral words are given a positive sentiment strength of 2 instead.
- A **booster word list** is used to strengthen (e.g., very +1) or weaken (e.g., somewhat -1) the sentiment of any immediately following sentiment words.
- A **negating word list** is used to neutralise any following sentiment words (skipping any intervening booster words). (e.g., “I do not hate him”, is not negative).
- An **emoticon list with polarities** is used to identify additional sentiment (e.g., :) scores +2).
- Sentences with **exclamation marks** have a minimum positive strength of 2, unless negative (e.g., "hello Pardeep!!!").
- **Repeated punctuation** with one or more exclamation marks boost the strength of the immediately preceding sentiment word by 1.

- **Two consecutive moderate or strong negative terms** with strength at least -3 increase the strength of the second word by 1 (e.g., "He is a nasty[-3] hateful[-4] person" scores -5 for negativity due to this boost.

There are also a number of additional rules in SentiStrength that have been tested but do not improve its performance. These are disabled in the default configuration but can be enabled by users if they are likely to work on a particular type of data.

- Sentiment terms in **CAPITAL letters** receive a strength increase bonus of 1.
- **Two consecutive moderate or strong positive terms** with strength at least 3 increase the strength of the second word by 1.
- **Sentences containing irony** have their positive sentiment reduced to 1 and their negative sentiment equal to 1 less than their positive sentiment. Irony is operationalized by the presence of a term or phrase from a user-defined list (e.g., politicians' names or derogatory terms for politicians).

Many of the additional rules can be disabled or modified in SentiStrength, if desired. For instance the booster words feature can be disabled by emptying the booster word list and the number of words allowed between a negating word and a sentiment word can be user defined. Caution should be used when modifying the defaults: whilst a change may improve scores on some texts it may reduce overall accuracy by giving worse scores on other, perhaps unexpected cases.

Some of the rules also need to be modified for non-English versions of SentiStrength and there are some options for this. For example, in Germanic languages negating words are typically placed after sentiment words and this aspect of the rule is a modification available in SentiStrength. If a test data set is used to evaluate SentiStrength then this test data should not also be used to evaluate SentiStrength rule modifications (or any other SentiStrength modifications) because this would invalidate the test results due to the potential for over-fitting the algorithm – i.e., tailoring it too much to the test data so that it is more accurate on the test data than on other similar data.

Supervised and Unsupervised Modes

SentiStrength has the capability to optimise its lexicon term weights for a specific set of human-coded texts (i.e., a collection of texts with human-assigned sentiment scores for each one). It does this by repeatedly increasing or decreasing the term weights by 1, one term at a time, and then assessing whether this change increases, decreases or does not affect the overall classification accuracy for the human coded texts. Changes that improve accuracy are kept and the process is repeated until no term strength change improves the overall classification accuracy (i.e., it is a hill climbing algorithm). This process can easily lead to *over-fitting* because only one occurrence of a term can be used to change its lexicon strength, although it is possible to increase the threshold required for a change in the algorithm. This means requiring a bigger increase in accuracy for a change in term strength in order to retain the change.

If the above process is used to optimise the SentiStrength weights then this is its *supervised* mode; without training is the *unsupervised* mode. As the results below show, supervised mode has similar overall accuracy to that of unsupervised mode, but it should logically outperform the unsupervised mode if large enough training data sets are used.

As a final point on the lexical term strength optimisation process, the reason why stemmed terms are included in the lexicon rather than a complete list of matching terms is to improve the power of the term strength optimisation algorithm because the stemmed terms can occur more often than each individual matching complete word.

Evaluating SentiStrength

SentiStrength can be evaluated by applying it to a set of texts that have been coded for sentiment by humans and comparing the SentiStrength scores with the human scores. For the best results, the average of at least three different human coders should be used for the texts. This is because coding is subjective and one coder is more likely to give unusual results than the average of three or more. The coders should be chosen and assessed for accuracy and consistency because a large number (>1000) of texts need to be coded for a reliable assessment. One way to select coders is to give them the same 100 texts to classify as a pilot study and then choose the three coders that agree most with each other. Experience with SentiStrength suggests that only 1 in 5 coders give accurate enough results to be useful. The best metric to assess the degree of agreement between the coders is Krippendorff's inter-coder weighted alpha (Krippendorff, 2004) with the weight for a mismatch being the difference between the two scores. This metric is one of the standard options for social sciences content analysis studies and is available as a menu option in the Windows version of SentiStrength. This metric can be used for the initial pilot and also to report the level of agreement on the complete data set for the selected coders once they have finished.

Once the human-coded corpus is ready, SentiStrength can be applied to it and its results compared with the human coder average. The best metric for the comparison is the Pearson correlation because this is one of the few standard performance metrics for sentiment analysis that takes into account how close an estimation is to the correct value when they are not identical. It is also superior in practice to the alternatives, such as Mean Absolute Deviation (MAD) and Mean Squared Error (MSE) in that it gives a result that is more easily interpreted by researchers outside the sentiment analysis field since the Pearson correlation is simple and well known.

The SentiStrength-human comparison gives two separate correlations, one for positive sentiment strength and one for negative sentiment strength. If these are significantly positive then this is evidence that SentiStrength works better than random guessing. Higher positive correlations indicate better performance and can be used to compare different versions or settings for SentiStrength and to compare its performance on different corpora, including those reported below.

Table 1. Unsupervised and supervised SentiStrength 2 against the baseline measure (predicting the most common class) and the standard machine learning algorithm (from a set of nine) and feature set size (from 100, 200 to 1000) having the highest correlation with the human-coded values. Correlation is the most important metric.

BBC Forums*	**Positive correct**	**Negative correct**	**Positive correlation**	**Negative correlation**
Unsupervised SentiStrength	51.3%	46.0%	0.296	**0.591**
Supervised SentiStrength	60.9%	48.4%	0.286	0.573
Best machine learning	**76.7%**	**51.1%**	**0.508**	0.519
Digg				
Unsupervised SentiStrength	53.9%	46.7%	0.352	0.552
Supervised SentiStrength	57.9%	50.5%	**0.380**	**0.569**
Best machine learning	**63.1%**	**55.2%**	0.339	0.498
MySpace				
Unsupervised SentiStrength	62.1%	70.9%	**0.647**	0.599
Supervised SentiStrength	62.1%	72.4%	0.625	**0.615**
Best machine learning	**63.0%**	**77.3%**	0.638	0.563
Runners World				
Unsupervised SentiStrength	53.5%	50.9%	0.567	0.541
Supervised SentiStrength	53.9%	55.8%	0.593	0.537
Best machine learning	**61.5%**	**65.3%**	**0.597**	**0.542**
Twitter				
Unsupervised SentiStrength	59.2%	66.1%	0.541	0.499
Supervised SentiStrength	63.7%	67.8%	0.548	0.480
Best machine learning	**70.7%**	**75.4%**	**0.615**	**0.519**
YouTube				
Unsupervised SentiStrength	44.3%	56.1%	0.589	0.521
Supervised SentiStrength	46.5%	57.8%	0.621	0.541
Best machine learning	**52.8%**	**64.3%**	**0.644**	**0.573**
All 6				
Unsupervised SentiStrength	53.5%	58.8%	0.556	0.565
Supervised SentiStrength	56.3%	61.7%	0.594	**0.573**
Best machine learning	**60.7%**	**64.3%**	**0.642**	0.547

***** The metrics used are: accuracy (% correct) and correlation. Best values on each data set and each metric are in bold. Source: extracted from (M. Thelwall et al., 2012)).

Table 1 reports the correlations between SentiStrength and the human coder average for a range of different types of social web text. The positive correlations in all cases together indicate its general applicability to social web texts, even in unsupervised mode. In other words it would be reasonable to apply it to any new source of social web texts, even in the absence of training data. The table shows that the supervised mode, with lexicon term weights automatically adjusted based upon the training data, is not clearly better overall than the unsupervised mode. Hence the advantage of creating human coded data for any new text source would be the ability to measure SentiStrength's performance rather than the ability to run it in supervised mode.

When evaluating supervised performance it is important to measure on a test set of texts that is different from the training set. The standard way of achieving this result is known as 10-fold cross validation and is used in Table 1 and is available as an option in SentiStrength. With this method a single set of human coded texts is used but is split into a training part (90% of the tests) and a testing part (the remaining 10% of the texts). This ensures that the training and test texts are different but does not give accurate results because only 10% of the texts are used for testing. To circumvent this accuracy issue the process is repeated for each remaining set of 10% of the texts and the 10 results are averaged to give a more precise accuracy estimate.

Sarcasm, Irony & Politics

The evaluation results in Table 1 contain two correlations that are lower than the rest (BBC Forums and Digg, both for positive sentiment strength only). An examination of the data revealed that many incorrect matches were associated with discussions of political and other controversial issues. These often employ irony and sarcasm in the form of ostensibly positive statements with a negative meaning, such as "warmongers will be happy that another 10 soldiers were killed today". In response to this problem a number of options to detect sarcasm were tested for SentiStrength but none were adopted because all incorrectly identified sarcasm more often than not. The most promising rule was that a text was sarcastic if it contained both positive and negative sentiment and a politician's name and a winky emotion ;) but even this rule failed. Sarcasm is known to be difficult to automatically detect (Gonzalez-Ibanez, Muresan, & Wacholder, 2011) and is often also problematic for humans, perhaps because its power is partly due to the cleverness with which it is constructed. There have been some small successes with automated sarcasm detection, however. Book reviews are one example due to the repeated use of stock sarcastic types of phrase, such as "this book has a great cover" that can be learned from a training corpus (Tsur, Davidov, & Rappoport, 2010). Sarcasm in Portuguese political discussions can also be identified through a combination of features including the use of a politician's name in diminutive form (Carvalho, Sarmento, Silva, & de Oliveira, 2009). These successes do not seem to transfer well to general sarcasm detection in English and so this seems to be a major challenge.

A consequence of the difficulty in detecting sarcasm and the problems that it causes is that SentiStrength is likely to have lower accuracy than normal for positive sentiment strength in sets of texts in which sarcasm is common, including political discussions. Whilst the results in Table 1 are still significantly positive and hence may be useful in practice, the performance is below human levels of accuracy.

Adaptations for Specific Topics

The typical sentiment of terms depends on the context in which they are used. For instance, in most contexts sentences containing any of the terms, "horror", "frightened", "scream" or "scared" would be negative but if the context is a horror movie review then these terms might tend to indicate a positive review instead (e.g., a good horror movie should be scary). Hence if SentiStrength is applied to texts from a relatively narrow context, such as a type of product review or discussions of a specific topic, then the lexicon may need to be modified to take into account the commonly used sentiment terms and strengths in the new context. This can be done manually with a human expert and a development corpus, perhaps also using common sense to reassess terms strengths as well as examining incorrect results on the development corpus. For instance if the development corpus suggests a new sentiment term then common sense may dictate that synonyms of that term should also be added.

SentiStrength also has a method to automatically suggest new terms based upon a development or training corpus. The *lexical extension method* proceeds as follows.

1. SentiStrength makes a list of all words in all texts in the data set and assigns each a score of zero.
2. All texts are classified by SentiStrength with its default lexicon.

3. For each text with a positive or negative score from SentiStrength different from the human coder score, a value is added to the score for each term in the text equal to the difference between the human coded and SentiStrength scores. This number is positive if SentiStrength is too high for the text and negative if it is too low.
4. SentiStrength prints out a sorted list of all terms with non-zero scores, sorted by the score value.
5. Either (automatic method) all terms with a sufficiently high or low score are added to the lexicon with a nominal weight (e.g., +3 or -3) or (manual method, recommended) a human expert scans the list and decides which terms to add and their strengths.

Although the normal term optimisation method used in supervised SentiStrength can adjust lexicon weights for existing terms, the lexicon extension method is able to identify new terms to add to the lexicon. The method has been tested on two corpora and gave small accuracy improvements in both cases (Table 2). Although the manual term adding variant did not give more accurate results than the automatic method the latter added some irrelevant terms, such as *letters* which is undesirable. Hence the manual method is preferred.

Table 2. Results of adding the topic-specific sentiment terms to the training and test corpora using two different methods. The highest values for each corpus are in bold. (source: (M. Thelwall & Buckley, in press)). The first two corpora used the single scale sentiment output and the other six used the dual scale output.

Corpus	Original correlation	Correlation with extra terms (1)	Correlation with extra terms (2)
Riots	0.4104	**0.4429**	0.4383
AV	0.4038	0.4124	**0.4126**

MySpace	0.5919(+)
	0.6023(-)
		0.6134(+)
		0.5963(-)
			0.6091(+)
			0.6041(-)

BBC	0.3357(+)
	0.6098(-)
		0.3376(+)
		0.6095(-)
			0.3376(+)
			0.6104(-)

Digg	**0.3566(+)**
	0.5709(-)
		0.3554(+)
		0.5715(-)
			0.3554(+)
			0.5709(-)

Runners World	**0.6318(+)**
	0.5632(-)
		0.6305(+)
		0.5632(-)
			0.6318(+)
			0.5632(-)

Twitter	**0.6027(+)**
	0.5160(-)
		0.6024(+)
		0.5160(-)
			0.6024(+)
			0.5160(-)

YouTube	**0.5933(+)**
	0.5498(-)
		0.5878(+)
		0.5461(-)
			0.5878(+)
			0.5462(-)

Mood Adaption

SentiStrength can be assigned either a positive or a negative mood. This mood determines the polarity of sentences that do not contain explicit sentiment polarity indicators but have an indication of the presence of sentiment, either through excessive punctuation (e.g., “look here!!) or through deliberate misspellings (e.g., “loooook”). In cases where energy (the term arousal is used by psychologists) is perceived without an indication of sentiment type humans seem to use contextual information to fill the gap (Fox, 2008). The original version of SentiStrength was developed for predominantly positive texts and had a fixed positive mood but the mood can be set to either positive or negative in the current version. To test which is better, both settings can be tried on a test or development set and the more accurate one retained. Setting the best mood can substantially improve performance (M. Thelwall & Buckley, in press).

Sentiment anomaly detection with local context

Any sentiment analysis program makes mistakes due to complex sentence constructions that it cannot decode (e.g., sarcasm). In some cases it may be possible to use contextual information to predict that a classification is likely to be incorrect and to automatically correct it without identifying the linguistic causes. This has been tested for SentiStrength with a simple rule to detect a sudden jump in either positive or negative sentiment. The rule is that if a sentiment score differs by over 1.5 from the average of the previous three contributions in the sequence then the new sentiment score is regarded as an anomaly and damped by adjusting it by bringing it 1 closer to the average of the previous three scores. This method has been shown to be capable of improving SentiStrength's accuracy in monologs (sequential lists of contributions by a single person, e.g., their tweet stream), dialogs (sequential lists of messages exchanged between two communication partners) and multi-participant discussions, but the improvement is only minor and depends upon the type and probably the topic of communication (M. Thelwall et al., 2013). For these reasons, more research is needed before this method is used outside of experimental settings.

Language Variants

SentiStrength can be customised for new languages by translating its sentiment lexicon and other resources, adjusting its optional settings to cope with language specific features, such as negating words occurring after sentiment terms, and refining through testing on a human-coded development corpus. For some languages, additional processing will be needed to get good results however. For example, the morphology of Turkish word formulation (Turkish is an agglutinative language) means that Turkish text must be parsed to separate out negating suffixes from sentiment terms and then the negation can be re-introduced by inserting an artificial negating word (e.g., _NOT_) prior to the sentiment word before being submitted to SentiStrength (Vural, Cambazoglu, Senkul, & Tokgoz, 2013). Languages without word boundaries in standard form also need to be first processed with an algorithm to split the characters into words. This applies to Chinese and Japanese. The pre-processing approach has also been employed by some commercial users for languages like French, Spanish and Portuguese where it is not strictly necessary in order to fit in with existing systems that work exclusively with lemmatised text. In this case all the SentiStrength resource files should also contain equivalently lemmatised text (i.e., with words parsed into standard abstract units known as lemmas).

Converting SentiStrength to work with a new language requires no coding in most cases because all its language resources are stored externally in plain text files and because it has language customisation options built in, including a UTF-8 mode for non-ASCII characters. Nevertheless a good conversion will take at least a month to translate the resource files, human-code a development corpus of 1000 texts and refine the lexicon and options based upon an examination of incorrect classifications in the development corpus. The accuracy of the translated variant should also be assessed before use on a second human coded corpus, to determine its accuracy. This is likely to be lower than SentiStrength's accuracy for English due to the longer development time for this language.

Application: The role of sentiment in Major Media events

To illustrate a simple application of SentiStrength to investigate sentiment patterns in social web texts, this section describes a case study of tweeting major events. Other chapters in this book illustrate applications of SentiStrength and other sentiment analysis programs to computing systems, modelling the spread or influence of sentiment online and psychological experiments to test it (M. Thelwall, Buckley, & Paltoglou, 2011).

The aims of the Twitter study were to identify the typical pattern of sentiment changes during a major event and to determine whether changes in sentiment could be used to predict the amount of interest in an event during the early stages of its evolution. The raw data used was a collection of 35 million tweets from February 9, 2010 to March 9, 2010. Major events were detected with a time series analysis of relative word frequencies in tweets (M. Thelwall & Prabowo, 2007), followed by human filtering. For each word a spike value was calculated - the biggest daily relative frequency increase in the proportion of tweets containing the word. For each day after the third the increase was calculated as the daily value minus the average of all previous daily values. The word list was then ranked in descending order of spike size and manually filtered to remove words describing the same event as a more highly-ranked word, words referring to purely online events (e.g., the Follow Friday hashtag #ff) and words associated with non-news events, such as Valentine's Day.

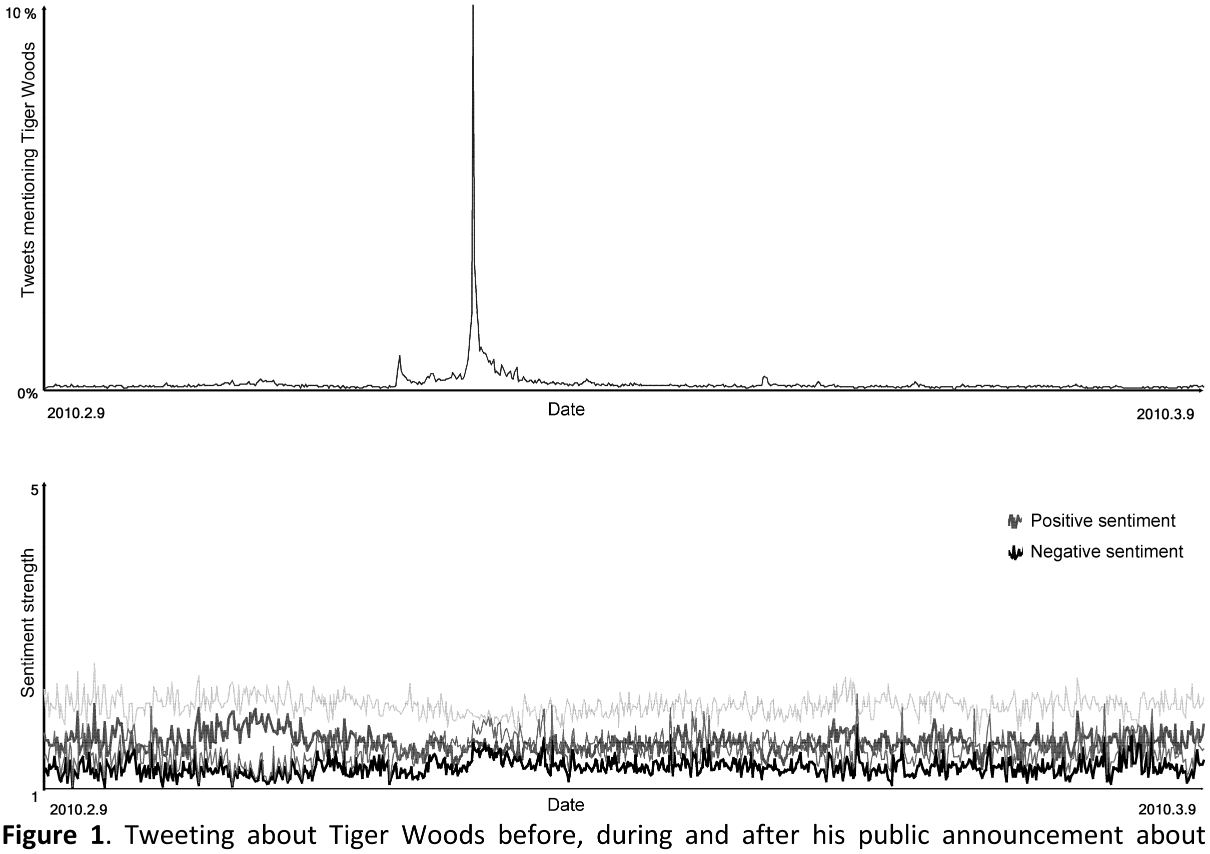

Figure 1. Tweeting about Tiger Woods before, during and after his public announcement about having an affair. The top graph shows the total volume of tweeting mentioning Tiger Woods. Black lines in the lower graph reveal average negative sentiment strength for all Tiger Woods tweets (thick line) and for subjective tweets (thin line); grey lines reflect positive sentiment.

For each of the top 30 remaining events, the sentiment scores of all matching tweets were calculated using SentiStrength and the average levels of sentiment before, during and after the event were calculated. Across all 30 events, the spikes were typically associated with small (average 6%) increases in negative sentiment but often no change in positive sentiment. Figure 1 shows a very negative event with only a small increase in average negative sentiment strength.

The main outcomes of the study were the result that sentiment changes were typically too small to be useful for predicting the importance of an event in its early stages, that negativity was the key sentiment for major media events in Twitter and that sentiment is rarely expressed explicitly in tweets about major events. This last fact is particularly surprising because major events presumably arouse strong emotions in order to trigger a sudden spike of tweeting so it seems that the sentiment is implicit in the sending of the tweet and does not need to be expressed.

Application: Sentiment in YouTube Comments

A second study examined the role of sentiment in YouTube comments; these are left by a small minority of viewers after or during a YouTube video and are interesting for the insights that they can give into viewer reactions to the video or its topic.

The study found that weak positivity was the most common sentiment in comments and this typically corresponded to mild praise for a video, its author or topic (M. Thelwall, Sud, & Vis, 2012). In addition, videos with stronger positive comments tended to have weaker negative comments and vice versa, suggesting a viewer consensus of opinions about the video. Some comments are replies to previous comments, however, and so any sentiments could be directed at other commenters rather than the video. Probably as a result of this, negativity significantly associated with the densest discussions within comment sections. In other words negativity was more successful than positively in fostering interactions, a phenomenon also found for other online contexts (Chmiel et al., 2011).

Conclusion

The chapter described the sentiment strength detection program SentiStrength that uses a dual positive/negative sentiment strength scoring system and is optimised for general social web text. SentiStrength employs a lexicon of sentiment words and word stems together with average positive or negative sentiment strength scores for them. Texts are classified with the largest positive or negative scores of any constituent word unless these are modified by any of the additional classification rules, such as in the case of emotions, negations and booster words.

SentiStrength has near-human accuracy on general short social web texts but is less accurate when the texts often contain sarcasm, as in the case of political discussions. The accuracy of SentiStrength can be enhanced by extending its lexicon and altering its mood setting for sets of texts with a narrow topic focus. As the case studies illustrate, SentiStrength can be used to analyse large scale sentiment patterns in the social web in addition to its commercial uses.

Acknowledgement

This work was supported by a European Union grant by the 7th Framework Programme, Theme 3: Science of complex systems for socially intelligent ICT. It is part of the CyberEmotions project (contract 231323).

References

Carvalho, P., Sarmento, L., Silva, M. J., & de Oliveira, E. (2009). Clues for detecting irony in user-generated contents: Oh...!! it's "so easy" ;-). *Proceedings of the 1st international CIKM workshop on topic-sentiment analysis for mass opinion (TSA '09)* (pp. 53-56). New York, NY: ACM Press.

Chmiel, A., Sienkiewicz, J., Thelwall, M., Paltoglou, G., Buckley, K., Kappas, A., & Hołyst, J. A. (2011). Collective emotions online and their influence on community life. *PLoS ONE, 6*(7), e22207.

Derks, D., Bos, A. E. R., & von Grumbkow, J. (2008). Emoticons and online message interpretation. *Social Science Computer Review, 26*(3), 379-388.

Fox, E. (2008). *Emotion science*. Basingstoke: Palgrave Macmillan.

Gonzalez-Ibanez, R., Muresan, S., & Wacholder, N. (2011). Identifying sarcasm in twitter: A closer look. *Proceedings of the 49th annual meeting of the association for computational linguistics* (pp. 581-586). Portland, Oregon: Association for Computational Linguistics.

Krippendorff, K. (2004). *Content analysis: An introduction to its methodology*. Thousand Oaks, CA: Sage.

Kucuktunc, O., Cambazoglu, B. B., Weber, I., & Ferhatosmanoglu, H. (2012). A large-scale sentiment analysis for yahoo! answers. Paper presented at the *Web Search and Data Mining (WSDM2012),* Seattle, Washington. 633-642.

Norman, G. J., Norris, C., Gollan, J., Ito, T., Hawkley, L., Larsen, J., . . . Berntson, G. G. (2011). Current emotion research in psychophysiology: The neurobiology of evaluative bivalence. *Emotion Review, 3*, 3349-359. doi: 10.1177/1754073911402403

Pennebaker, J., Mehl, M., & Niederhoffer, K. (2003). Psychological aspects of natural language use: Our words, our selves. *Annual Review of Psychology, 54*, 547-577.

Stone, P. J., Dunphy, D. C., Smith, M. S., & Ogilvie, D. M. (1966). *The general inquirer: A computer approach to content analysis*. Cambridge, MA: The MIT Press.

Thelwall, M., & Buckley, K. (in press). Topic-based sentiment analysis for the social web: The role of mood and issue-related words. *Journal of the American Society for Information Science and Technology,*

Thelwall, M., Buckley, K., Paltoglou, G., Skowron, M., Garcia, D., Gobron, S., . . . Holyst, J. A. (2013). Damping sentiment analysis in online communication: Discussions, monologs and dialogs. In A. Gelbukh (Ed.), *CICLing 2013, part II, LNCS 7817* [null] (pp. 1-12). Heidelberg: Springer.

Thelwall, M., Buckley, K., & Paltoglou, G. (2011). Sentiment in twitter events. *Journal of the American Society for Information Science and Technology, 62*(2), 406-418.

Thelwall, M., Buckley, K., & Paltoglou, G. (2012). Sentiment strength detection for the social web. *Journal of the American Society for Information Science and Technology, 63*(1), 163-173.

Thelwall, M., & Prabowo, R. (2007). Identifying and characterising public science-related concerns from RSS feeds. *Journal of the American Society for Information Science & Technology, 58*(3), 379-390.

Thelwall, M., Sud, P., & Vis, F. (2012). Commenting on YouTube videos: From guatemalan rock to el big bang. *Journal of the American Society for Information Science and Technology, 63*(3), 616–629.

Tsur, O., Davidov, D., & Rappoport, A. (2010). ICWSM - A great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In W. W.

Cohen, & S. Gosling (Eds.), *Proceedings of the fourth international AAAI conference on weblogs and social media* (pp. 162-169). Washington, D.C.: The AAAI Press.

Vural, G., Cambazoglu, B. B., Senkul, P., & Tokgoz, O. (2013). A framework for sentiment analysis in turkish: Application to polarity detection of movie reviews in turkish. In E. Gelenbe, & R. Lent (Eds.), *Computer and information sciences III: 27th international symposium on computer and information sciences* [null] (pp. 437-445)

Walther, J., & Parks, M. (2002). Cues filtered out, cues filtered in: Computer-mediated communication and relationships. In M. Knapp, J. Daly & G. Miller (Eds.), *The handbook of interpersonal communication (3rd ed.)* (pp. 529-563). Thousand Oaks, CA: Sage.

Weber, I., Ukkonen, A., & Gionis, A. (2012). Answers, not links: Extracting tips from yahoo! answers to address how-to web queries. *Proceedings of the fifth ACM international conference on web search and data mining (WSDM '12)* (pp. 613-622). New York: ACM Press.