A Lexicon-based Unsupervised Approach to Recognizing Changes in Opinions - Case Study on 'Artificial Intelligence'

Ruben Blom
10684980

Bachelor thesis
Credits: 18 EC

Bachelor Opleiding Kunstmatige Intelligentie

University of Amsterdam
Faculty of Science
Science Park 904
1098 XH Amsterdam

Supervisor
Dr. A. Bilgin

Institute for Language and Logic
Faculty of Science
University of Amsterdam
Science Park 904
1098 XH Amsterdam

June 24th, 2016

Abstract

In recent years the interest in public opinion mining through sentiment analysis has increased rapidly. Sentiment analysis aims to calculate the semantic orientation of documents. This orientation is often represented by a numeric value or *positive*, *negative* and *neutral*. This thesis focuses on the analysis of public opinion from news and magazine articles. A general framework analyzing various corpora preprocessing and analyzing documents is proposed. Unsupervised lexicon-based approaches are discussed and compared to supervised machine-learning sentiment-analysis toolboxes. For a case study, documents related to "Artificial Intelligence" are collected using keyword search and analyzed for sentiment orientation using the proposed framework. The resulting data is then visualized in a data plot. Furthermore, historical events related to AI are gathered, and the change in sentiment for the documents for each event is calculated using linear regression and visualized in different subplots. It is shown that the framework is capable, given the keywords, of identifying the change in public opinion automatically.

Contents

1 Introduction

The analysis of public opinion is traditionally the research conducted by journalists, social scientists and businesses that specialize in opinion polling. However, in recent years, the mining of public opinion has shifted to algorithmic approaches, and the interest in public opinion mining algorithms has increased rapidly. With the expansion of publicly available information sources like Twitter and Facebook –and the increase in their users– more companies and governments have become interested in analyzing public opinion about them and their products through sentiment analysis. Sentiment analysis is the automatic identification of the writer's attitude with respect to a given topic(Pang and Lee, 2008). This attitude is often represented through a sentiment polarity: a number between -1 and 1, where -1 is a *strong negative* attitude, 0 is *neutral* and 1 displays a *strong positive* one (Pang and Lee, 2008). Other methods use scores for *positivity*, *negativity* and *objectivity* such that together they add up to 1 (Esuli and Sebastiani, 2006).

The research on how opinions change over time is often referred to as opinion tracking (Ku et al., 2006). It aims to illustrate supportiveness towards an entity or topic by drawing its degree along a timeline. These techniques are often utilized by organizations to survey customer happiness about their products or to improve them (Dave et al., 2003). Opinion tracking is applied in other contexts as well: to monitor opinions on certain matters –political bills, for example– and how the public views on these matters change over time.

This work focuses on the analysis of topical opinions from news articles and magazines. As a case study, the topic 'Artificial Intelligence' will be used. The goal of this work is threefold: firstly, to develop a sentiment classification system that can be used to process an article, such that the document-level sentiment polarity can be classified. Secondly, a visualization system is to be developed that is able to plot the polarity scores against the document publication dates in order to properly analyze the development of public opinion on the topic over time. Lastly, using the visualization, more in-depth research is to be conducted on the topic 'Artificial Intelligence'. Is there a change in the sentiment of the documents following historical events concerning the topic?

In order to achieve these goals the following question is addressed:

> *How did public opinion on Artificial Intelligence' evolve through time?*

To answer this question, news articles and magazines from various sources will be gathered and bundled as corpora. Since Artificial Intelligence (AI) consists of different subfields (Russell et al., 1995), multiple keywords will be used searching for these documents. Furthermore, this work explores various methods for preprocessing the raw text, compares different sentiment lexicons and toolkits by testing them on annotated datasets, and will try to discover how public opinion on AI changed over time by analyzing major events of AI.

This thesis consists of five parts. In Section 2, the existing literature will be reviewed. Next, in Section 3, a general framework for preprocessing and analyzing documents will be proposed. The experiments and results are presented in Section 4, and a conclusion is given in Section 5. Lastly, the results and the conclusion will be discussed, and future work will be proposed in Section 6.

2 Literature Review

2.1 Sentiment Analysis

The term sentiment analysis is said to be first used by Yi et al. (2003) and opinion mining by Dave et al. (2003). Although the term *opinion* is much broader than *sentiment*, they are most often used interchangeably in the field of study. As mentioned in the introduction, the task of sentiment analysis is to predict the writer's attitude by analyzing text documents. Using sentiment-bearing words or expressions in those documents, sentiment analysis aims to calculate the semantic orientation (positive, negative or neutral) of them (Pang and Lee, 2008). Semantic orientation is also referred to as the *polarity*. Consequently, analyzing documents concerning a particular topic helps to determine the opinion on this subject (Ku et al., 2006).

The field of study has two research directions: Sentiment Analysis and Subjectivity Classification. Pang et al. (2002) and Turney (2002) pioneered the first by classifying a large number of opinions into a positive and negative category by using machine learning techniques. Wiebe (2000) used Subjectivity Classification by classifying text documents into the subjective or objective class.

For extracting the sentiment, there are two main approaches: a lexicon-based approach and a machine-learning approach (Liu, 2012). Furthermore, sentiment analysis can be performed on three different levels: word-level, sentence-level and document-level (Liu, 2012)s. Lexicon-based approaches use dictionary lookups to assign each feature (one of more word tokens) a prior polarity score. Other features can manipulate this score; a process which is described later. The result is a word-level score. The final scores for the features are then aggregated into a single score for the sentence-level score. By then aggregating the sentence scores, the document-level score is calculated. This way of calculating sentiment is based on two assumptions: individual words have a *prior polarity*, and this is independent of context. These assumptions follow the research of Osgood et al. (1964), and several approaches have used these assumptions (Wiebe, 2000; Hu and Liu, 2004; Kim and Hovy, 2004). To create such dictionaries various methods have been researched. In a manual approach, Cesarano et al. (2004) asked human subjects to score opinion-expressing documents on which statistical methods were applied to derive the score for each word. Using WordNet(Fellbaum, 1998) – a dictionary that lists synonyms and antonyms for a given English word – Kamps et al. (2004) used the relative distance between the terms *good* and *bad* to determine the sentiment score. A corpus-based approach was proposed by Hatzivassiloglou and McKeown (1997): using a corpus, seed words with a known polarity and a set of linguistic rules, additional sentiment adjectives can be found in the corpus. For example, the conjunction word AND can be applied to the sentence, "These flowers smell delicious and sweet." If "delicious" is a seed word, known to be positive, then "sweet" will also be positive. Using this and other connectives like OR, BUT, EITHER, other sentiment-bearing adjectives can be found (Liu, 2012).

Machine-learning approaches use a statistical approach to building classifiers that are trained on a dataset using features like unigrams or bigrams (sets with one or two words) with or without Part-of-Speech tags (see section 2.3.1). Pang et al. (2002) experimented with various features and machine-learning algorithms. They showed that training Support Vector Machines using unigrams without Part-of-Speech tags seemed to be the most successful approach in this category. The problem with the machine-learning approach, however, appears to be that they perform well solely in the domain that they are trained on (Pang and Lee, 2008).

2.2 Aggregation from Word to Sentence and Document-level

Because lexicon-based approaches return a word-level sentiment polarity, the results have to be aggregated into a sentence-level sentiment. In order to calculate this sentence-level score Muhammad et al. (2013) used an aggregate-and-average strategy described by the following formula:

$$S_p = \frac{1}{n}\sum_{i=1}^{n} Sw_i$$

where S_p and S_w are sentiment scores of sentence p and word w and n is the total amount of words in p. The same strategy was applied to traverse from sentence to document level and is described by the following formula:

$$S_d = \frac{1}{m}\sum_{j=1}^{m} Sp_j$$

where S_d and S_p are the sentiment scores of document d and sentence p and m is the number of sentences in document d.

2.3 Features Used in Sentiment Analysis

2.3.1 Part-of-Speech

A Part-of-Speech (POS) is a set of lexical items that have the same grammatical properties. Common Parts-of-Speech in the English language are *noun*, *verb*, *adjective* and *adverb*. POS information is often used in sentiment analysis because using POS-tags can help distinguish the sense of a word by way of disambiguation (Wilks and Stevenson, 1998). For example, the information that "cold" is a verb makes it disjoint from the noun "cold", referring to the "common cold". A list of five frequently found POS-tags from the Penn Treebank Project (Marcus et al., 1994) is shown in Table 1.

Adjectives have been used as features for sentiment analysis by a number of researchers for both lexicon-based and machine-learning-based approaches (Mullen and Collier, 2004; Whitelaw et al., 2005). In their research, Hatzivassiloglou and McKeown (1997) found that the presence of adjectives has a high correlation towards sentence subjectivity. Often, this research is used as evidence that (certain) adjectives are good predictors of sentiment polarity. Turney (2002), however, proposed that, rather than isolating the adjectives, document sentiment could be detected based on selecting phrases, which contained a number of pre-specified POS patterns, most including an adjective or an adverb. In a study by Pang et al. (2002) that performed sentiment classification on movie reviews, it was found that using only adjectives performed much worse than using an equal amount of most frequent unigrams. They pointed out that verbs and nouns also can be strong indicators of sentiment.

2.3.2 Negation

Another import concern in the field of sentiment analysis is handing negations. While the sentences "I like this movie" and "I don't like this movie" may appear similar to a machine, the negation token "n't" changes the sentiment from "positive" into "negative". When a lexical term is negated, the most obvious approach is to flip the sign of the sentiment score (Saurí, 2008). If "like" has a score of +0.8 then "don't like" will get a score of -0.8. This is often referred to as *switch negation*. However, not all negation terms

Tag	Description
JJ	Adjective
NN	Noun
NNP	Proper noun
RB	Adverb
VB	Verb

Table 1: Five frequently used POS-tags

reverse the polarity of a lexical term. For example, the adjective "excellent" is considered to have a strongly positive sentiment. Its negated version "not excellent", however, is far from a strongly negative sentiment-bearing term. *Shift negation* (Machova and Marhefka, 2014) captures this phenomenon better by shifting the polarity instead of reversing it. In order to account for the negation, a sentiment score is shifted with a fixed amount towards the opposite polarity. For example, if excellent" has a score of +1, then "not excellent" shifts towards the opposite, which is negative polarity, with a score of say, -0.8. The result is that "not excellent" receives a sentiment score of 0.2, which captures the intended meaning of the phrase better than the switch negation. Pragmatically, polarity shifts reflect the reality of negations better. This claim is supported by Horn, who suggested in his book *"A natural history of negation"* (Horn, 1989) the non-symmetrical relation between affirmative and negative sentences.

2.3.3 Negativity Weighting

Kennedy and Inkpen (2006) found that lexicon-based sentiment analysis has the tendency to be positively biased. Voll and Taboada (2007) shifted the numerical cut-off point where reviews were classified as positive or negative to overcome this problem. A different approach was implemented by Taboada et al. (2011) where the final semantic orientation of any negative score was increased by 50%. Their intuition was that giving negative expressions more weight was theoretically more satisfying.

2.4 Publicly Available Lexicons

There are various lexicons publicly available that map words to a form of semantic orientation. A selection of these lexicons is discussed below.

2.4.1 General Inquirer

The General Inquirer is a widely used lexical resource compiled by Stone et al. (1966). It features a "Positiv" or "Negativ" category for words with a positive or negative outlook. For the former, the Inquirer contains 1915 words and for the latter 2291, for a total of 4206 words. For example, "accept" is part of the Positiv set and "agony" is tagged as Negativ. The General Inquirer, however, only uses three POS-tags: "Noun", "Modif" and "SUPV". The "Modif" tag captures sentiment modifiers like adjectives and adverbs, while "SUPV" indicated a verb. Occasionally, a comment is present in the lexicon to further specify the tag and the use of the word.

2.4.2 MPQA Subjectivity Lexicon

The Subjectivity Lexicon (Wilson et al., 2005) is a collection containing 8222 opinion words that are tagged with a positive, neutral or negative prior polarity. The lexicon has an extra feature, type, which distinguishes between the weak subjective words and strongly subjective ones. For example, the adjective "above-average" is tagged as "weakly positive" and the verb "abuse" as "strongly negative".

2.4.3 SentiWordNet

The SentiWordNet lexicon (Esuli and Sebastiani, 2006) was created using WordNet (explained in section 2.1) and features the same synonym sets (synsets), sets that contain words sharing the same meaning. SentiWordNet (SWN) has two main versions, which are 1.0 and 3.0. Both associate a synset `s` with two numerical scores, `Pos(s)` and `Neg(s)`, ranging from 0 to 1. A `lemma#POS` pair can have more than one sense. For example, the noun "cold" can refer to the "common cold" and to "low temperature". Looking up a `lemma#pos` pair results in all WordNet entries for that pair sorted by their use frequency and their scores. In Table 2 the entries for the noun "cold" is shown as an example. Different senses can have different polarities, and a lexicon-based approach has no way of knowing which sense is the right one in the source text. Therefore, two general methods are used by researchers: use the most common one (first entry) or use the average of all senses. Guerini et al. (2013) researched these approaches and found that SWN version 3.0 is better than SWN version 1.0 and that selecting just one sense performs significantly less than using the average score.

POS	Offset	Pos(s)	Neg(s)	SynsetTerms
n	14145501	0	0.125	`cold#n#1`
n	05015117	0	0.125	`cold#n#2`
n	05725676	0	0	`cold#n#3`

Table 2: SentiWordNet entries for `cold#n`Daarn

2.4.4 SO-CAL

The Semantic Orientation CALculator (Taboada et al., 2011) consists of various dictionaries of words associating them with their polarity and strength ranging from -5 to 5 and it incorporates intensification and negation. SO-CAL has a dictionary with 1550 nouns, 2827 adjectives, 1142 verbs and 876 adverbs, totaling 6395 words. Additionally, it has a dictionary with 219 intensifiers that are used to amplify or downtone, i.e. respectively increase or decrease the sentiment orientation. SO-CAL has intensifiers of different lengths; they range from one to four words. For example, "to a certain extent" is one of the four-word intensifiers from the dictionary. The intensity of the modifier is stored as a percentage of change. For example: "slightly" modifies a lexical term by multiplying it by -50%, "very" increases a term with 25%. In SO-CAL, intensifiers can be applied recursively starting from the closest one to the word. Amplifying "good", which has a value of 3, with "really very" results in $(3 \times [100\% + 25\%] \times [100\% + 15\%]) = 4.3$ for "really very good".

2.4.5 Summary

In Figure 3 the number of POS-tagged words are summarized. For the General Inquirer, the combined JJ and RB cell are the amount of words tagged as "Modif".

Lexicon	Adjectives	Adverbs	Nouns	Verbs
Inquirer	2775		5148	2728
MPQA	3250	330	2170	1325
SentiWordNet	18156	3621	82115	13767
SO-CAL	2827	876	1550	1142

Table 3: Amount of words in the lexicons, split by POS-tag.

2.5 Existing Sentiment Analysis Toolboxes

2.5.1 Stanford CoreNLP

Stanford CoreNLP (Manning et al., 2014), developed by Stanford University, is a toolkit with various tools for natural language processing. CoreNLP[1] splits a source text into sentences and tokens and annotates them with the appropriate tools. These tools include a tokenizer, POS-tagger, a named entity recognizer (NER), a lemmatizer a sentiment analyzer and a basic dependency parser. A tokenizer is a function that splits sentences into word and punctuation tokens; the POS-tagger returns all the POS-tags for the tokens in a sentence and the NER annotators marks named entities including names, locations and time (Nadeau and Sekine, 2007). A lemmatizer is a lexical tool that returns the lemmas for all given words. Lemmatization is the process of converting a word by stemming it and grouping different inflected forms of it. For example, "walking" becomes "walk" and "better" is converted to "good" Plisson et al. (2004). The sentiment annotator returns the sentence-level sentiments in the form of polarity scores. The basic dependency parser shows relations in a sentence like negations and modifiers. An example of a parsed sentence is provided in Figure 1.

[1] http://stanfordnlp.github.io/CoreNLP/

2.5.2 AlchemyAPI

AlchemyAPI[2] is an IBM company that uses machine learning techniques that were developed using deep learning to offer services in Natural Language Processing and Computer Vision. The technology used for these services is similar to IBM's Watson computer (Williams, 2013). AlchemyAPI includes a commercial parser that can extract features from plain text or can pull information from a web page by providing HTML page cleaning (removing HTML tags) and using the extracted raw text as input. Extracted features include named entities, keywords, document-level sentiment. For the named entities and keywords the relevance of it to the source text is calculated, and the sentiment of the entity or keyword is shown, see Figure 2. While AlchemyAPI has a free plan that offers 1000 credits per day, each extracted feature costs one credit, which makes it less attractive for using on big corpora.

2.6 Data Acquisition

To collect data on the case study 'Artificial Intelligence', multiple resources have been used. First, keywords related to AI were collected. The corpora were searched for these keywords, and the matched documents were stored in the database for further processing.

[2]http://www.alchemyapi.com/
[3]http://www.bloomberg.com/news/articles/2013-08-13/florida-to-sue-georgia-in-u-s-supreme-court-over-water

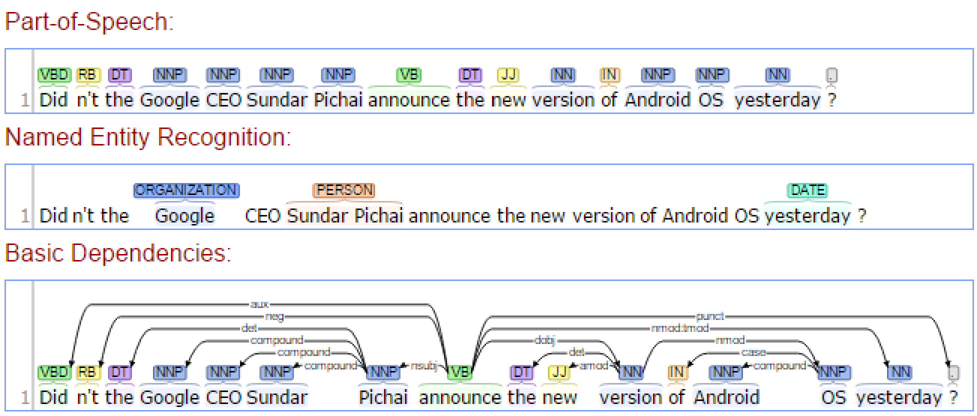

Figure 1: Example of parsing a sentence using Stanford CoreNLP

Entity	Relevance	Sentiment	Type	Subtypes	Linked Data
Florida	0.88536	mixed	StateOrCounty	Location PoliticalDistrict AdministrativeDivision GovernmentalJurisdiction USState	dbpedia freebase yago website
Georgia	0.694097	mixed	StateOrCounty		
Rick Scott	0.659522	negative	Person		dbpedia freebase
Apalachicola Bay	0.531865	negative	GeographicFeature	Location BodyOfWater	dbpedia freebase yago
U.S. Supreme Court	0.458005	negative	Organization	GovernmentalBody	dbpedia freebase yago website

Figure 2: AlchemyAPI's NER analysis on the Bloomsberg's online article.[3]

2.6.1 Keyword Collection

The keywords related to AI were collected using the site AITopics[4] as a reference. AITopics has the largest collection of information about the research and application of AI related topics. The gathered keywords are displayed in Table 4.

A. Newell	autonomous robot	natural language processing
A. Turing	autonomous vehicle	neural network
Alan Turing	autonomous weapon	pattern recognition
Allen Newell	data mining	reinforcement learning
H. Simon	decision tree learning	semantic web
Herbert A. Simon	deep learning	speech recognition
M. Minsky	expert system	speech synthesis
M.I.T	face recognition	statistical learning
Marvin Minsky	fuzzy logic	strong ai
Russel Norvig	intelligent agent	turing test
Turing Award	knowledge engineering	weak ai
artificial intelligence	machine learning	robot
autonomous agent	multi-agent	

Table 4: Used keywords for the search on AI related documents.

[4]http://www.aitopics.org

2.6.2 Unannotated Resources

Corpus of Historical American English The Corpus of Historical American English (COHA) is the largest structured corpus of historical English (Davies, 2010). It contains more than 400 million words of text between the 1810s-2000s. The COHA is split up into different categories: news and magazines articles, and, non-fiction and fiction books. For this research, the search for AI related articles is limited to news and magazine articles because those are relatively short and expected to be more opinionated. The result of using the keywords for searching the COHA is shown in Table 5, the first two columns. COHA consists of text files marked with a year and an identifier. The text file contains text from a news- or magazine article from that date; however, it doesn't specify the publication month or day. In Figures 3 and 4 the amount of documents that were found per year is displayed.

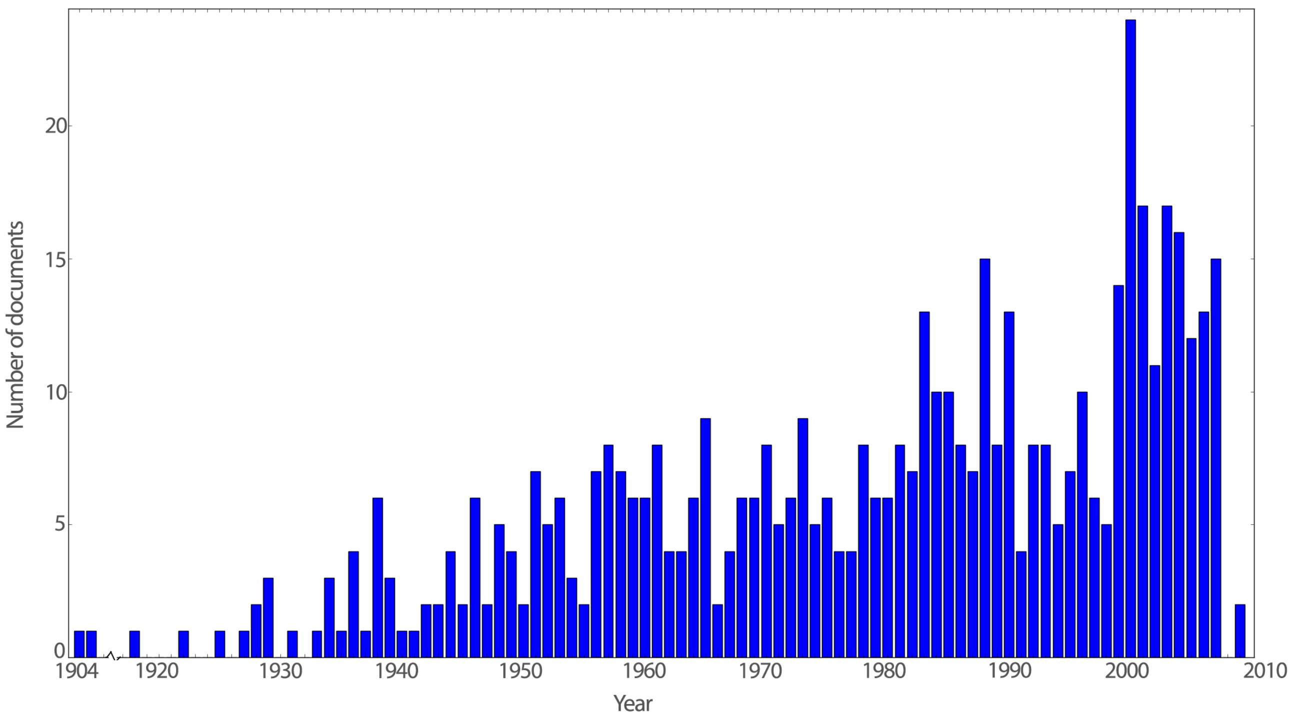

Figure 3: Amount of data per year for magazine articles.

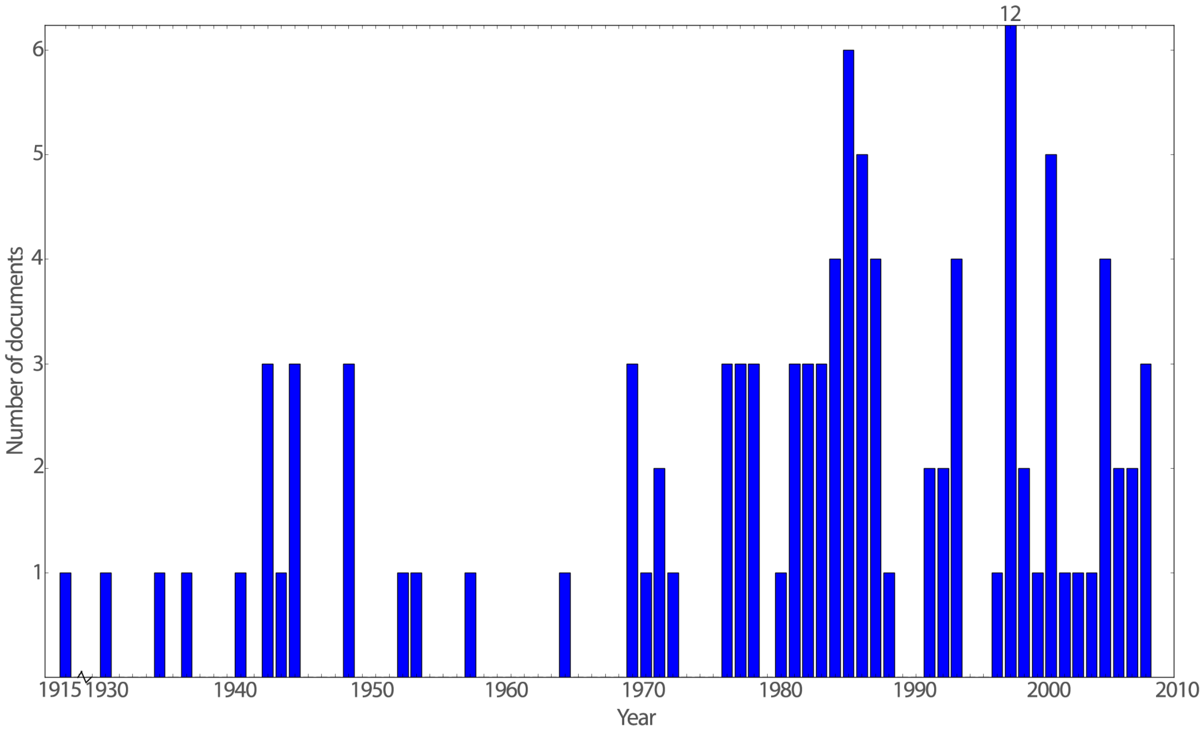

Figure 4: Amount of data per year for news articles.

TIME Vault TIME Vault[5] is the online archive of the TIME Magazine containing every issue since 1923. It offers users the ability to search the magazine articles and read them as HTML text. By searching for the keywords and scraping the found articles pages, a corpus was created. The amount of found document for each keyword is displayed in Table 5, the last column. A downside is that most older articles are not available without subscribing. Most acquired documents, therefore, have a date range from the 2000s-2010s, see Figure 5.

[5]http://www.time.com/vault/

keyword	News COHA	Magazines COHA	TIME Vault
“A. Turing”	0	0	4
“Alan Turing”	0	2	43
“artificial intelligence”	11	32	45
“autonomous agent”	0	2	1
“autonomous robot”	0	2	10
“autonomous vehicle”	0	1	16
“autonomous weapon”	0	0	2
“data mining”	1	4	46
“deep learning”	1	3	11
“expert systems”	0	3	0
“face recognition”	0	2	32
“fuzzy logic”	0	3	4
“H. Simon”	2	4	6
“Herbert A. Simon”	0	0	1
“intelligent agent”	1	2	3
“M.I.T”	17	167	44
“machine learning”	0	0	47
“Marvin Minsky”	0	0	5
“multi-agent”	0	0	1
“natural language processing”	0	0	12
“neural network”	0	10	17
“pattern recognition”	0	4	29
“semantic web”	0	0	3
“speech recognition”	0	3	31
“speech synthesis”	0	1	1
“statistical learning”	0	0	1
“strong ai”	5	6	0
“turing test”	2	2	16
“weak ai”	2	2	0
robot	70	322	45
Total keyword hits:	112	577	476

Table 5: Results of searching documents using the keywords on all unannotated resources.

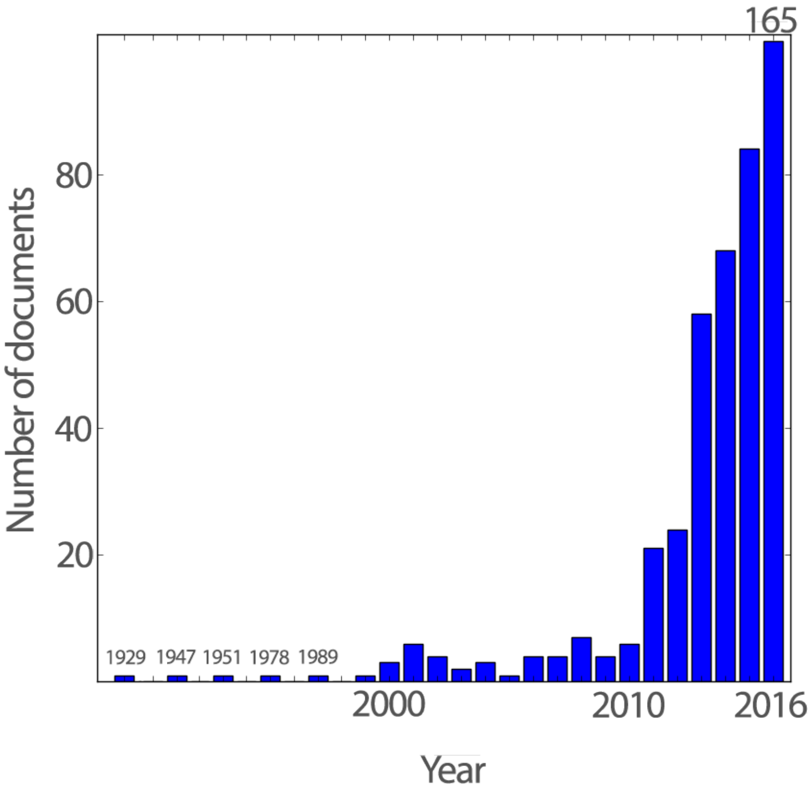

Figure 5: Amount of data per year for the TIME Vault.

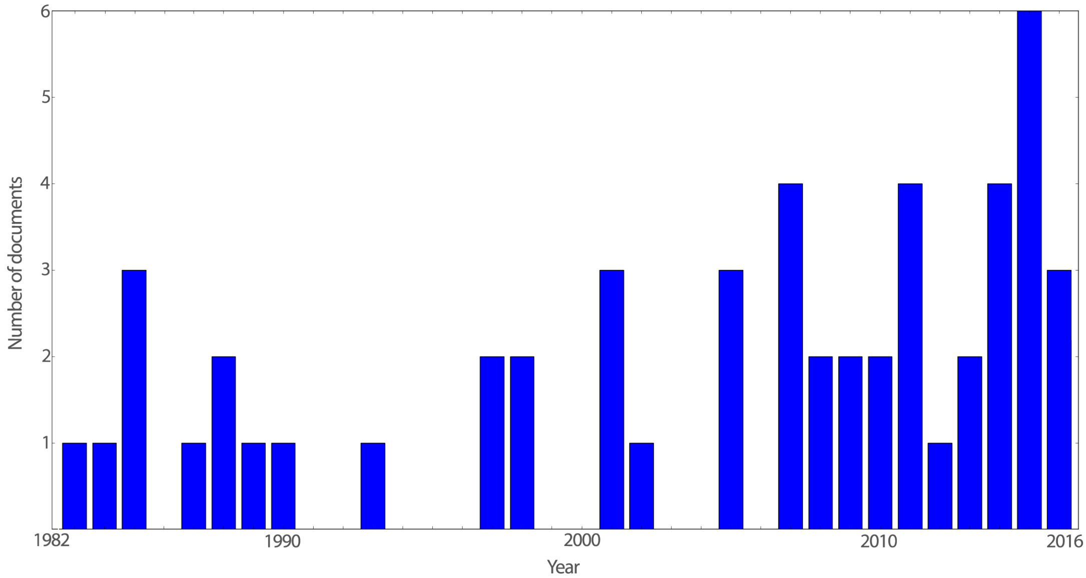

Figure 6: Amount of data per year for the New York Times corpus.

New York Times The New York Times[6], or the *Times*, is an American daily newspaper that also has international importance. Besides the daily news, the Times also covers Technology and Science. Additionally, it offers a section for opinion columns. The Times offers a developer API, which can be used to find articles based on a search string. Although the Times offers subscribers access to older editions of the newspaper, those editions are scanned versions and the article text is not extracted. If a search string matches an item from the old editions, the API only returns an excerpt of it. The New York Times corpus consists only of documents that were found by using the keyword 'artificial intelligence' and 52 documents were returned using this keyword. The amount of documents per year is displayed in Figure 6.

2.6.3 Annotated Resources

Movie review corpus For a fair comparison of the unsupervised lexicon-based approaches and the sentiment analysis toolkits, the movie review data from Pang et al. (2002) their experiments was used. It is found that sentiment analysis on movie reviews is more complicated than analysis on other domains (Chaovalit and Zhou, 2005; Thet et al., 2010; Turney, 2002). This is due to the fact that the positive movie reviews often contain unpleasant scenes, and the negative ones often include mentioning of pleasant scenes (Turney, 2002). Because of the corpus being challenging

[6]http://www.nytimes.com/

	name	# documents	# words	min. year	max. year
Unannotated	News COHA	108	14462	1904	2009
	Magazine COHA	528	138420	1915	2007
	TIME Vault	471	7	1929	2016
	New York Times	52	8	1981	2016
Annotated	Pos. Movie Reviews	1000	59135	n.a.	n.a.
	Neg. Movie Reviews	1000	56583	n.a.	n.a.

Table 6: Summary of all corpora and their statistics.

for sentiment analysis, it was chosen as a comparison method for the algorithms. This corpus consists of 1000 positive and 1000 negative movie reviews. The reviewers marked the reviews with either a *thumbs up* or a *thumbs down* on writing the review, marking it as being either positive or negative sentimental oriented.

2.6.4 Corpora Summary

In Table 6, the corpora are summarized. The number of documents and words are shown, along with the range of years of publication.

2.6.5 AI Related Events

year(range)	event
1943	End WW2
1950	Turing Test & I, Robot
1956	Top-Down AI
1966	Heavy blow for NLP
1968	A Space Odyssey
1974-1980	AI Winter
1981	Expert Systems
1990	Bottom-Up AI
1996	First Deep Blue Win
2008	Speech Recognition

Table 7: Gathered historical events on AI.

To explain the changes in the sentiment data, important events related to AI were gathered from various from timelines from the BBC[7] and Wikipedia[8]. They are displayed in Table 7.

[7] www.bbc.co.uk/timelines/zq376fr

[8] https://en.wikipedia.org/wiki/Timeline_of_artificial_intelligence

Some events need further highlighting:

- In 1950, Isaac Asimov published his boot "I, Robot" along with the Three Laws of Robots. A book that is still a famous science fiction and has been made into a movie in 2004. In that same year, Alan Turing published the "Turing Test". A proposed method for determining intelligence in a computer.
- In 1956, the term 'Artificial Intelligence' was created at the Dartmouth College conference. Marvin Minsky, an influential academic at that time, favored a top-down approach. Start tackling AI by looking at the rules that govern human behavior. The US government gave Minsky substantial funding, hoping it would give them the upper hand in the Cold War.
- In 1966, negative reports on machine translation killed much work in Natural language processing done in the past ten years.
- Influenced by Minsky in 1968, the film "2001: A Space Odyssey" was published, featuring an intelligence computer, the HAL 9000.
- The AI Winter was a result of strong criticism from the US Congress. Millions had been spent, and little results had been published to show for it. Funding for the industry was cut, and the AI Winter was the result.
- In 1981, AI's commercial value rose again, attracting new investments. Instead of focusing on creating general intelligence, the focus was on much smaller tasks. The "Expert Systems" needed to be programmed with only the rules for that particular problem. Based on this approach, the first commercial product, RI, was developed. A program that helped to configure new orders for computer systems.
- Expert systems couldn't solve the problem of imitating the brain. In 1990, Rodney Brooks published a new paper: "Elephants Don't Play Chess". He helped the revival of Bottom-Up AI, resulting in the reboot of the long-unfashionable field of neural networks.
- In 1996, IBM's top-down machine beat his first professional chess opponent. And in 1997, Deep Blue beat chess champion, Garry Kasparov.
- In 2008, Google released its speech recognition app on the Apple iPhone. Before then, speech recognition never performed above 80% accuracy. Google's new approach used thousands of powerful computers learning to spot patterns in the human speech.

3 Approach

To answer the research question, a timeline will be created with the date on the x-axis and the sentiment polarities on the y-axis. To preprocess the documents, analyze the documents' sentiment polarity and to plot the data, a general framework was proposed. The architecture of this framework is displayed in Figure 7 and is discussed in this Section.

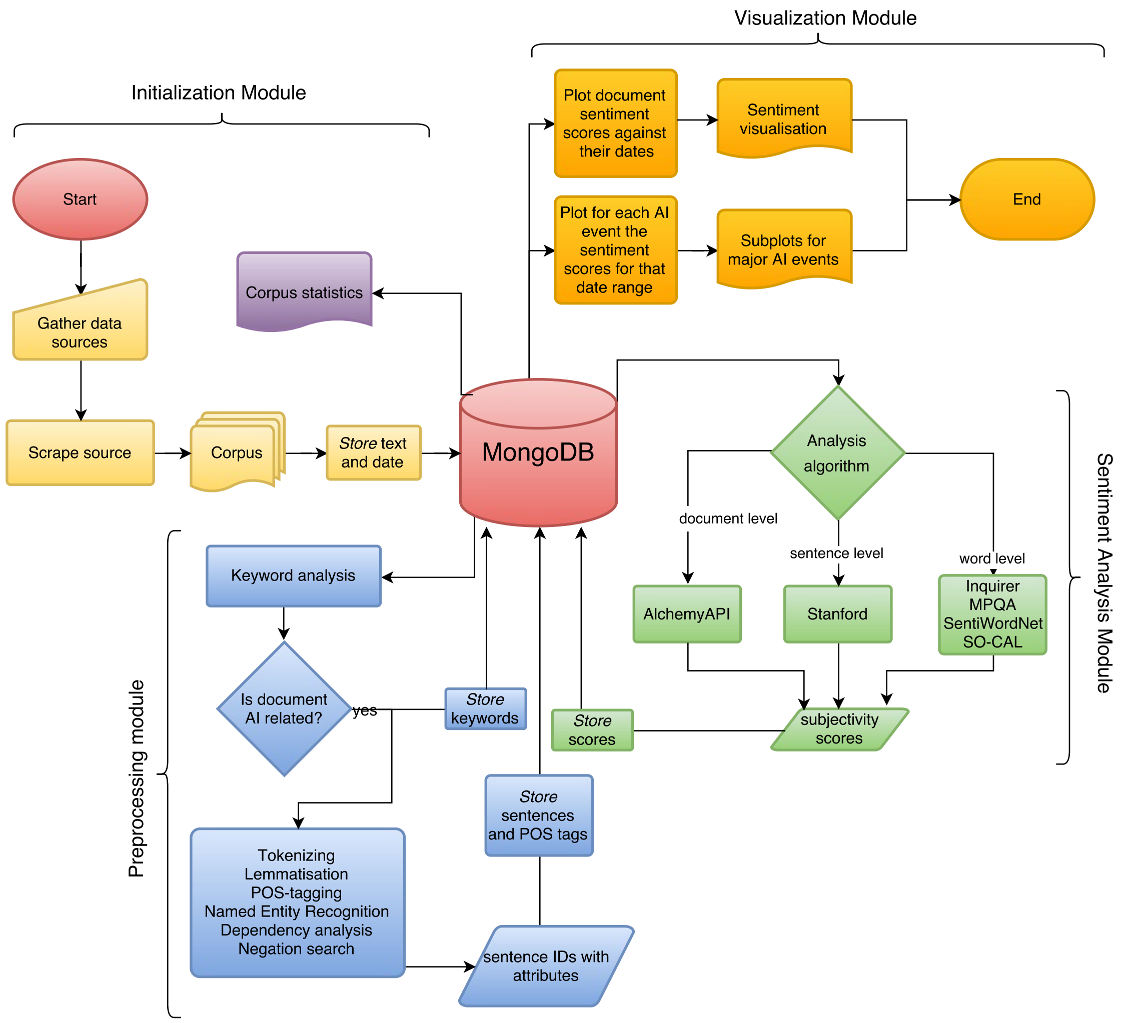

Figure 7: The architecture of the proposed general framework.

3.1 Architecture of the General Framework

The proposed framework contains four different modules. The first module is the Initialization Module that can process a data source, extract the dates and upload them to the database. If the data source, like the TIME Vault, is a website, the module can perform a keyword search, scrape the returned results for the date and raw text and uploads the result to the database. The Initialization Module is discussed in Section 3.2. The second module, discussed in Section 3.3, is the Preprocessing Module. It uses the CoreNLP toolkit to preprocess documents and append the results to the document entry. The third module called the Sentiment Analysis Module, can perform sentiment analysis on preprocessed document using the six discussed algorithms. This module is discussed in Section 3.4. The fourth and last module, discussed in Section 3.5, is the Visualization Module. It retrieves the data from the database and plots it one timeline and for each case study-related event it plots a zoomed in subplot.

3.2 Initialization Module and Data Storage

The gathered data, discussed in Section 2.6, is stored in the database through the Initialization Module. If a corpus consists of a number of text files combined in a folder, the module will try to find the publication dates in the filename. The text contained in the files is then transferred to the database.

The module can also find and scrape web pages if a corpus is on-line. The date and article text is then extracted from the web page and stored in the database. The database used in this research is MongoDB[9]. The various corpora are stored as `collections`, which contain the `documents`. By keeping the corpora separated, the user can choose which corpora to use in their analysis and the statistics for each corpus can be easily extracted.

3.3 Preprocessing Module

In order to use lexicon-based sentiment analysis on the documents, each document in that particular corpus needs to be preprocessed, and its features need to be extracted. For this task, Stanford's CoreNLP was utilized. CoreNLP can be run as a server that accepts calls with text and a list of annotators. It parses the text into sentences and annotates them with various attributes. The result is returned in JSON format and can be read in Python as a dictionary, mapping every annotator's name to its results. In this research, the annotators `lemma`, `pos`, `ner` and `depparse` were used. The annotator `pos` tags every word in the sentence with its POS-tag. `ner`, marks all named entities and `depparse` finds dependencies between words. They

[9] https://www.mongodb.com/

have been discussed in Section 2.5.1. All documents are preprocessed, and their results are stored using the following algorithm:

```
Data: A list of collections
Result: Updated preprocessed documents
foreach collection in list of collections do
    foreach document in collection do
        Send the document text to the CoreNLP API ;
        Retreive JSON result ;
        Read result as Python dictionary ;
        foreach sentence in result do
            Store sentence features under sentence index ;
            foreach dependency in sentence do
                if dependency is a negation then
                    Store negation under word index of sentence;
                end
            end
        end
        Update document with its sentences and their extracted
         features ;
    end
end
```

Algorithm 1: Preprocess documents in collections.

In section 2.3.2 the importance of negation search was explained. In CoreNLP's dependency parse negations are returned as (token index, negated word) pair. In the database, negations are stored as a boolean "True" for that word's index when a word is negated. An example of the results after preprocessing a document is found in the Appendix in Figure 11.

3.4 Sentiment Analysis Module

In the framework, each lexicon and toolbox are implemented as a method that returns the sentence-level scores `sentence polarities` and document-level score `sentiment` by using a retrieved `document`. For AlchemyAPI, the document text is sent to the server and the document-level sentiment is returned. AlchemyAPI doesn't offer sentence-level sentiments. Stanford CoreNLP's `sentiment` annotator returns the sentiment of each sentence in the text from the full document text. The document-level sentiment is calculated by taking the average of all sentence scores.

For the lexicon-based approaches, the aggregate-and-average strategy, as discussed in Section 2.2, was applied to calculate the sentence and document-level scores.

In these approaches, the word tokens from the tokenizer and their POS-tags from the POS-tagger are used to find the prior polarity for each of these words for each lexicon. As mentioned in Section 2.4.4, the SO-CAL lexicon includes an intensifier dictionary that is recursively applied. The longest intensifier has four words, so in the SO-CAL implementation the multiplier for the prior polarity is calculated as follows:

Input: current word, intensifiers
Result: intensifier multiplier
Initialize multiplier = 1;
Initialize wordlist = word preceding the current word in reverse order
;
for *index = size wordlist to 0* **do**
  if *wordlist[index-4: index] is an intensifier* **then**
    multiplier = multiplier * found intensifier-block
  end
  else if *wordlist[index-3: index] is an intensifier* **then**
    multiplier = multiplier * found intensifier
  end
  else if *wordlist[index-2: index] is an intensifier* **then**
    multiplier = multiplier * found intensifier
  end
  else if *wordlist[index] is an intensifier* **then**
    multiplier = multiplier * found intensifier
  end
end

Algorithm 2: Algorithm to calculate the intensifier for a given word.

3.5 Visualization Module

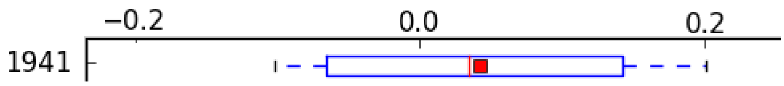

Figure 8: Example boxplot for 1941, turned 90 degrees for better visualization.

In order to show the results of the sentiment analysis and be able to analyze them across dates, a timeline is to be created on which they are plotted. Since COHA and NYT corpus data, which take up the majority of the documents, do not have a full date but only the publication *year*, this is the resolution the data will have to be showed on. Because of this, the data with the same year will be plotted in boxplots on the x-axis, resulting in a clear picture of the spread of the data within that year. An example of a boxplot of all data for the year 1941 is shown in Figure 8. The average score across all documents for that year is shown as a red square. The boxplot shows the range of the sentiments, the first percentile, median and the third percentile.

The data was gathered by iterating over the data, described by the following algorithm:

```
Initialize empty dictionary ;
foreach collection in collections do
    foreach document in dollection do
        Add sentiment score to scorelist at document year in
          dictionary
    end
end
foreach Year in dictionary do
    Plot data for that year at x-coordinate year number
end
```

Discretized Events To create a better view on the sentiment changes caused by the events, another method of plotting the data was implemented. This was implemented by discretizing the sentiment data around the events with a window range from `year` to `year+2` and plotting the data in different subplots. The resulting data points are then averaged to create the red squares similar to the plot over all documents. Over these data points, linear regression is then applied to calculate a degree of change by calculating the slope of the resulting line. The result of this method on the AI data is discussed in the results.

4 Experimentations and results

For unsupervised approaches to sentiment analysis, several lexicons for analyzing text on sentiment polarity have been discussed in Section 2. In this section, first, the lexicons will be compared by analyzing the annotated Movie Review corpus. Based on these results and factors like their limitations an algorithm will be chosen for analyzing the corpora on AI. In addition, the results given by the existing toolboxes will be presented to compare their performance.

The two machine-learning and four lexicon-based approaches have been tested on the positive and negative set from the Movie Review corpus to get a sense of their performance before other features are applied. For the lexicon-based approaches, only words with the POS-tags `JJ` (adjective) and `RB` (adverbs) are taken into account as these are major sentiment-bearing words, as discussed in Section 2.3.1. The accuracy of the approaches is shown in Table 8. Both machine-learning approaches have a clear bias towards the negative. Also, CoreNLP's performance on the positive Movie Review corpus was so low that a random algorithm would have performed better. Regrettably, AlchemyAPI, although scoring the highest on average, is a commercial service which makes it less suitable for this research. Therefore, for the further course of this research, the three best scoring lexicon-based approaches, SentiWordNet, MPQA and SO-CAL will be used. These approaches are subject to improvement when more features are added, and they also perform more consistent across various domains.

4.1 Negation

If a word is marked as being negated, the prior polarity is modified. For negation, two methods have been discussed in Section 2.3.2. Both shift negation and switch negation have been implemented in each lexicon's algorithm, and the accuracy scores are compared in Table 3 and 4. Because the results are almost equivalent, it was decided to use the shift negations for it follows the intuition better.

	Inquirer	SentiWordNet	MPQA	SO-CAL	AlchemyAPI	CoreNLP
pos	0.863	0.570	0.844	**0.894**	0.61	0.21
neg	0.343	0.740	0.525	0.480	**0.98**	0.99
avg	0.603	0.655	0.685	0.687	**0.80**	0.60

Table 8: Baseline analysis on all lexicon-based and machine-learning approaches. Pos: Positive Movie Reviews, neg: Negative Movie Reviews, avg: average.

	SentiWordNet	MPQA	SO-CAL
pos	0.832	0.838	**0.885**
neg	0.457	**0.552**	0.504
avg	0.645	**0.695**	**0.695**

Table 9: Accuracy of different lexicons using switch negation.

	SentiWordNet	MPQA	SO-CAL
pos	0.843	0.844	**0.893**
neg	0.437	**0.540**	0.483
avg	0.640	**0.692**	0.688

Table 10: Accuracy of different lexicons using shift negation.

4.2 Usage of POS-tags

To further investigate the assumption that adverbs and adjectives are good indicators for the document polarity, each lexicon-based algorithm has an option to use or ignore a word based on its POS-tag. Ignored words are skipped and are not taken into account when calculating the sentence-level polarity. The results of this experiment are presented in Table 11, Table 12 and Table 13 which show their accuracy.

4.3 Negative Weighting

As discussed in section 2.3.3, if a score is negative after all modifiers have been applied, it is multiplied by 150% to overcome the bias towards positivity. The results are displayed in Tables 14 through 16. SentiWordNet does not seem to bear the negative weighting well, while MPQA and SO-CAL improved by its implementation with the latter having the highest accuracy. Furthermore, the results show that using adjectives and adverbs *only* results in the highest accuracy score, confirming the theory that they are good sentiment bearers. Therefore, SO-CAL using adjectives and adverbs was applied for the analysis of all documents for the final results of this research.

4.4 Visualization Results

The visualization of all documents, analyzed and grouped per year using boxplots is shown in figure 9. All discussed AI events are annotated with a caption. It can be seen that all yearly scores are not far away from the neutral score, 0. It is possible that this is an effect caused by the averaging of the data from word-level to the score for each year. Although the scores vary not much form a neutral score, the figure shows that historical events of AI (e.g. the Turing Test) have influenced the sentiment polarity of the documents in the resources.

	SentiWordNet	MPQA	SO-CAL
pos	0.827	0.849	**0.933**
neg	0.465	**0.540**	0.411
avg	0.646	**0.695**	0.672

Table 11: Accuracy of different lexicons using adjectives and adverbs only.

	SentiWordNet	MPQA	SO-CAL
pos	0.816	0.846	**0.921**
neg	0.469	**0.495**	0.454
avg	0.643	0.671	**0.687**

Table 12: Accuracy of different lexicons using adjectives, adverbs and verbs.

	SentiWordNet	MPQA	SO-CAL
pos	0.823	0.526	**0.893**
neg	0.442	**0.615**	0.483
avg	0.633	0.571	**0.688**

Table 13: Accuracy of different lexicons using adjectives, adverbs, verbs and nouns.

	SentiWordNet	MPQA	SO-CAL
pos	0.383	0.656	**0.819**
neg	**0.847**	0.788	0.662
avg	0.615	0.722	**0.741**

Table 14: Accuracy of different lexicons using negative weighting on adjectives and adverbs.

	SentiWordNet	MPQA	SO-CAL
pos	0.491	0.633	**0.799**
neg	0.775	**0.800**	0.680
avg	0.633	0.717	**0.740**

Table 15: Accuracy of different lexicons using negative weighting on adjectives, adverbs and verbs.

	SentiWordNet	MPQA	SO-CAL
pos	0.469	0.519	**0.779**
neg	0.772	**0.854**	0.665
avg	0.621	0.687	**0.722**

Table 16: Accuracy of different lexicons using negative weighting on adjectives, adverbs, verbs and nouns.

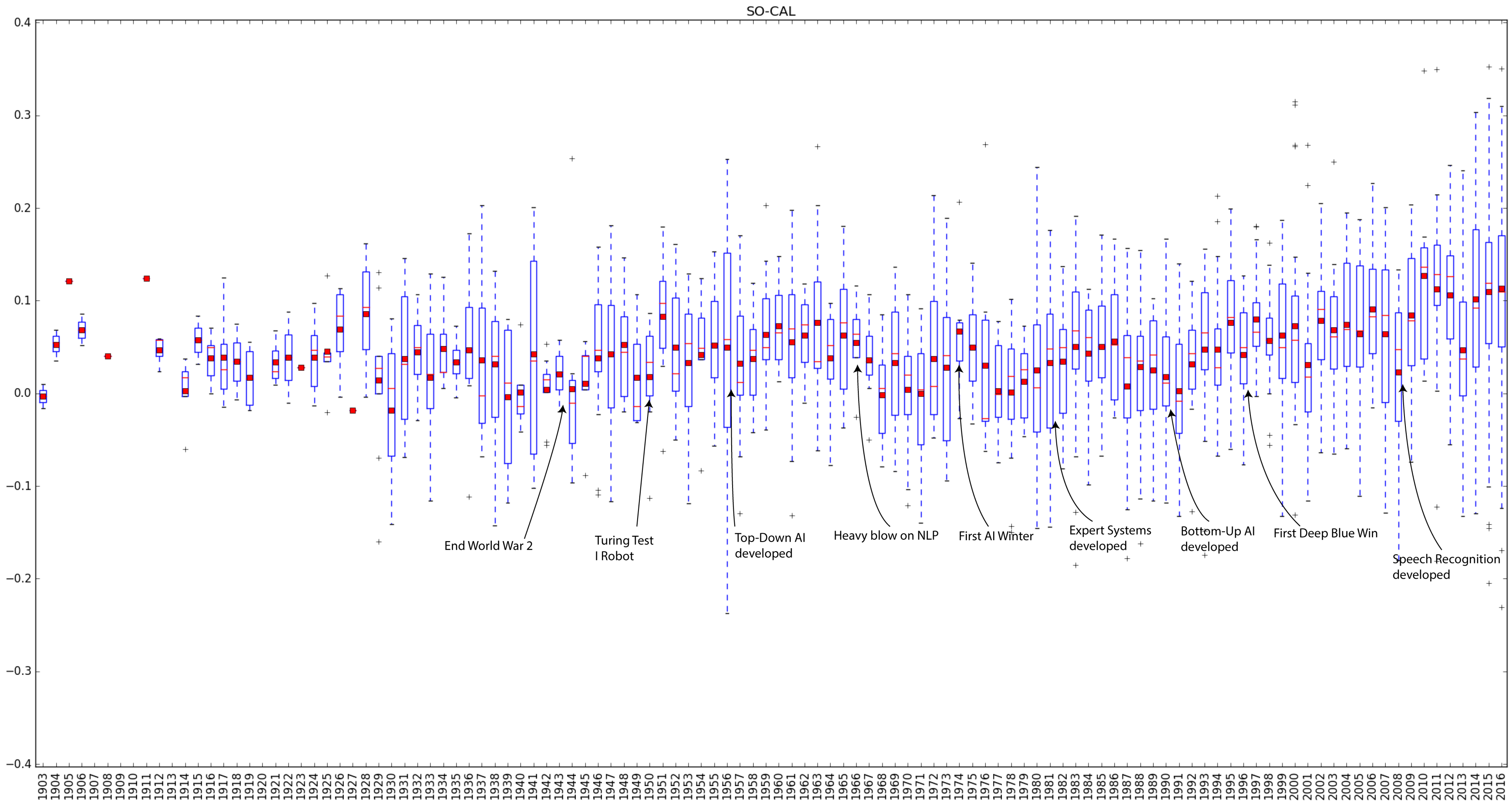

Figure 9: All documents grouped per year in boxplots captioned with AI events.

4.4.1 Discretized Events

In figure 10, the discussed events in the history of AI are shown after discretization on their year range. The major AI events clearly resulted in a change in sentiment of the documents: the end of World War 2 created a boost in science and the sentiment on AI clearly increases. The publication of the Turing Test and Isaac Asimov's "I, Robot", created a rise in sentiment for the documents as well. The blow on NLP by the ALPAC report created in a great decline in sentiment for the two years after and the AI Winter is clearly shown as well. While the AI Winter is reported to take place between 1974 and 1980, it can be seen that the sentiment rises towards the end of that period. Another big event in AI was the development of speech recognition software from Google. Although the data is becoming noisier at that period, as can be seen in the full plot in Figure 9, due to the increasing coverage on AI related news, the data clearly makes a huge leap towards the positive polarity.

Smaller events in AI appear to be harder to detect in the sentiment data. The start of a Top-Down approach on AI shows a slow decline in overall document sentiments. One possibility is that because others favored Bottom-Up AI at that time, it creates a stalemate, preventing to show a clear change in sentiment or the events weren't covered enough by the resources.

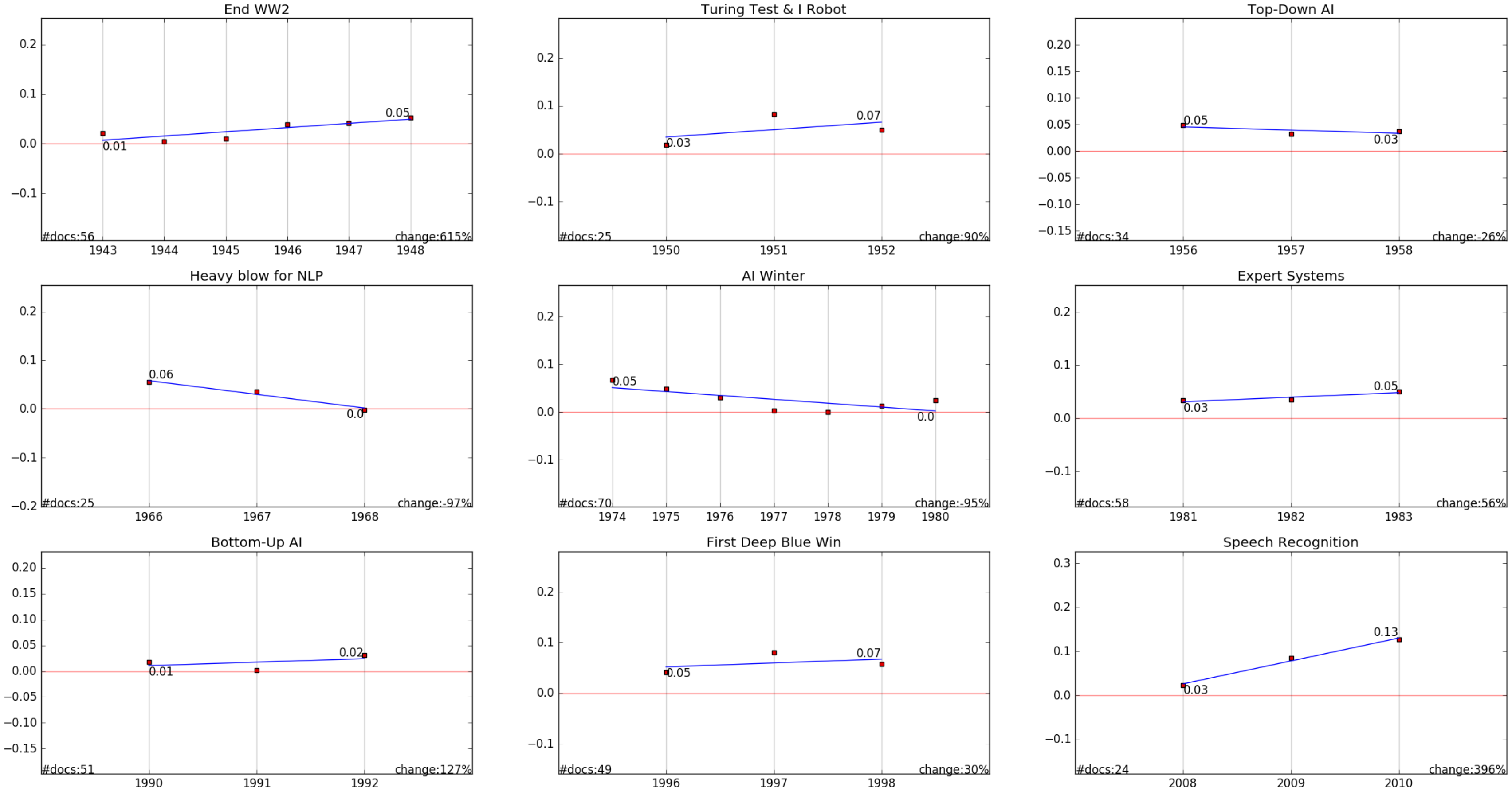

Figure 10: AI events in a discrete time frame, plotted with a linear regression to show change.

5 Conclusion

A general framework for analyzing various corpora using an unsupervised lexicon-based approach using the lexicons “General Inquirer”, “MPQA”, “SentiWordNet” and “SO-CAL” has been proposed. These approaches were compared to supervised sentiment analysis toolkits “CoreNLP” and “AlchemyAPI”. These approaches were compared by performing sentiment classification on the Movie Reviews corpus, a challenging task for sentiment analysis and thus a fair comparison method. The baseline results showed that almost all approaches performed equally except for AlchemyAPI. Further improvement of the lexicon-based approaches shows that it was possible to increase the performance of the best scoring lexicon, SO-CAL, from an average accuracy of 0.687 to 0.741 by using only adjectives and adverbs. This result also confirms the literature mentioning that adjectives and adverbs are good sentiment polarity indicators. MPQA and SO-CAL, the two highest performing lexicons are also the smallest, suggesting that the choice of which words to include in a lexicon is more important than how many words are in them.

For the case study on “Artificial Intelligence”, documents related to AI were searched based on keywords. The found documents were then preprocessed, combined as corpora and analyzed for their sentiment scores. The sentiment scores were combined into a document-level score by using an aggregate-and-average strategy and the resulting data was visualized. Because the used corpora only had a publication year available, and no month or day, scores were plotted on a yearly basis by aggregating the scores of documents per year into an overall-year-score. The resulting data for the period 1903-2016 was visualized using type of plots and regression techniques. Analysis on these visualizations has shown that major events in AI indeed reflected on the TIME magazine, New York Times and articles from the COHA, from which public opinion about AI is extracted. Hence, it is shown that the framework is capable, given the keywords, of identifying the common sense of events in the history automatically.

6 Discussion and Future Work

One of the assumptions underlying the framework presented here is that the accuracy obtained by the unsupervised lexicon-based approaches on the Movie Review corpus also carries to other corpora. While it is shown in the literature that this corpus contains a domain that is complicated for sentiment analysis, the actual performance cannot be verified. Therefore, further research on the validity of the results on the corpora is recommended. Another assumption lies in the strategy that calculates the document-level based on the word sentiment orientations. While averaging the data can be a good method to summarize data, in the approach taken the data was averaged from word- to sentence-level, from sentence-level to document-level and from document-level to a overall-year-score. Averaging results in losing subtle parts of the data, and it can be seen that the yearly scores do not vary a lot from the null line. Hence, researching other ways to aggregate from word-level to a yearly score is recommended for future work.

Appendix


```
{
    "_id" : ObjectId(),
    "texthash" : "2fe7a97f2c166fad",
    "sentiment" : {
        "AlchemyAPI: "0.520049", StanfordNLP": "-0.4245954654",
        "SentiWordNet" : -0.0231481481, MPQA" : -0.3324074074,
        "SO-CAL" : -0.120370370
    },
    "text" : "LAST winter, the satirical comedy troupe that calls itself
    Artificial Intelligence took over the Ballroom, ...",
    "sentence polarities" : {
        "SentiWordNet" : {"1" : 0.0, "2" : -0.333333333333333, ... },
        "MPQA" : {"1": 0.0, "2": ... }
        ...
    },
    "date" : "1987",
    "sentences" : {
        "0" : {"1" : "LAST", "2" : "winter", ...},
        "1" : {...},
        ...
    }
    "lemmas" : {
        "0" : { "1" : "last", "2" : "winter", "3" : ... },
        "1" : {...},
    ...
    },
    "ner" : {
        "0" : {"1" : "DATE", "2" : "DATE", ... },
        "1" : {...},
        ...
    },
    "pos" : {
        "0" : {"1" : "NN", "2" : "NN", .... },
        "1" : {...},
        ...
    },
    "negations" :{
         "0" : {"20": True}
    }
}
```


Figure 11: Example document entry taken from the NYT corpus.

References

AlchemyAPI (2014). Alchemyapi. http://www.alchemyapi.com/. Accessed: Jun 21, 2016.

Cesarano, C., Dorr, B., Picariello, A., Reforgiato, D., Sagoff, A., and Subrahmanian, V. (2004). Oasys: An opinion analysis system. In *AAAI Spring Symposium on Computational Approaches to Analyzing Weblogs (CAAW 2006)*, pages 21–26.

Chaovalit, P. and Zhou, L. (2005). Movie review mining: A comparison between supervised and unsupervised classification approaches. In *Proceedings of the 38th annual Hawaii international conference on system sciences*, pages 112c–112c. IEEE.

Dave, K., Lawrence, S., and Pennock, D. M. (2003). Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In *Proceedings of the 12th international conference on World Wide Web*, pages 519–528. ACM.

Davies, M. (2010). The corpus of historical american english. http://corpus.byu.edu/coha/.

Esuli, A. and Sebastiani, F. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, volume 6, pages 417–422. Citeseer.

Fellbaum, C. (1998). *WordNet*. Wiley Online Library.

Guerini, M., Gatti, L., and Turchi, M. (2013). Sentiment analysis: How to derive prior polarities from sentiwordnet. *arXiv preprint arXiv:1309.5843*.

Hatzivassiloglou, V. and McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. In *Proceedings of the 35th annual meeting of the association for computational linguistics and eighth conference of the european chapter of the association for computational linguistics*, pages 174–181. Association for Computational Linguistics.

Horn, L. (1989). A natural history of negation.

Hu, M. and Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM.

Kamps, J., Marx, M., Mokken, R. J., Rijke, M. d., et al. (2004). Using wordnet to measure semantic orientations of adjectives.

Kennedy, A. and Inkpen, D. (2006). Sentiment classification of movie reviews using contextual valence shifters. *Computational intelligence*, 22(2):110–125.

Kim, S.-M. and Hovy, E. (2004). Determining the sentiment of opinions. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1367. Association for Computational Linguistics.

Ku, L.-W., Liang, Y.-T., Chen, H.-H., et al. (2006). Opinion extraction, summarization and tracking in news and blog corpora. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 100107.

Liu, B. (2012). Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167.

Machova, K. and Marhefka, L. (2014). Opinion classification in conversational content using n-grams. In *Recent developments in computational collective intelligence*, pages 177–186. Springer.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J., and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Marcus, M., Kim, G., Marcinkiewicz, M. A., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., and Schasberger, B. (1994). The penn treebank: Annotating predicate argument structure. In *Proceedings of the Workshop on Human Language Technology*, HLT '94, pages 114–119, Stroudsburg, PA, USA. Association for Computational Linguistics.

Muhammad, A., Wiratunga, N., Lothian, R., and Glassey, R. (2013). Domain-based lexicon enhancement for sentiment analysis.

Mullen, T. and Collier, N. (2004). Sentiment analysis using support vector machines with diverse information sources. In *EMNLP*, volume 4, pages 412–418.

Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Osgood, C. E., Suci, G. J., and Tannenbaum, P. H. (1964). *The measurement of meaning*. University of Illinois Press.

Pang, B. and Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135.

Pang, B., Lee, L., and Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics.

Plisson, J., Lavrac, N., Mladenic, D., et al. (2004). A rule based approach to word lemmatization. In *Proceedings C of the 7th International Multi-Conference Information Society IS 2004*, volume 1, pages 83–86. Citeseer.

Russell, S., Norvig, P., and Intelligence, A. (1995). A modern approach. *Artificial Intelligence. Prentice-Hall, Egnlewood Cliffs*, 25:vii.

Saurí, R. (2008). *A factuality profiler for eventualities in text.* ProQuest.

Stone, P. J., Dunphy, D. C., and Smith, M. S. (1966). The general inquirer: A computer approach to content analysis.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., and Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307.

Thet, T. T., Na, J.-C., and Khoo, C. S. (2010). Aspect-based sentiment analysis of movie reviews on discussion boards. *Journal of information science*, page 0165551510388123.

Turney, P. D. (2002). Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics.

Voll, K. and Taboada, M. (2007). Not all words are created equal: Extracting semantic orientation as a function of adjective relevance. In *Australasian Joint Conference on Artificial Intelligence*, pages 337–346. Springer.

Whitelaw, C., Garg, N., and Argamon, S. (2005). Using appraisal groups for sentiment analysis. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 625–631. ACM.

Wiebe, J. (2000). Learning subjective adjectives from corpora. In *AAAI/IAAI*, pages 735–740.

Wilks, Y. and Stevenson, M. (1998). The grammar of sense: Using part-of-speech tags as a first step in semantic disambiguation. *Natural Language Engineering*, 4(02):135–143.

Williams, A. (2013). Alchemyapi raises $2 million for neural net analysis tech, on par with IBM Watson, Google. https://techcrunch.com/2013/02/07/alchemy-api-raises-2-million-for-neural-net-analysis-tech-on-par-with-ibm-watson-google. Accessed: Jun 20, 2016.

Wilson, T., Wiebe, J., and Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics.

Yi, J., Nasukawa, T., Bunescu, R., and Niblack, W. (2003). Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 427–434. IEEE.