Project: Search Engines and Information Retrieval Systems

Jens Arvidsson <jensarv@gmail.com>
Fredrick Chahine <fchahine@kth.se>
Erik Hallström <erik_hallstrom@icloud.com>
Petter Salminen <petsal@kth.se>

May 16, 2013

Contents

Abstract

In the following project we made a program that performs *Sentiment Analysis* for a given brand name based on one of three input sources: a live recording using the microphone, a folder containing several audio files, or a string inserted into the text box.

Our client, Jussi from Gavagai, wants us to investigate the possibility of developing a platform for Sentiment Analysis for brands given input data in the form of digital audio.

Using an `API` provided to us by Jussi, the program grades the given input on the five given levels: Sexy, Violence, Positive, Negative and Uncertainty.

For example, the given input of "When I drink coca cola I feel super good and it makes me smile so much yeah coca cola is my favorite drink" tells us that Coca-Cola in this sentence is seen in a positive light.

When the speech-to-text `API` works great, our program works like a charm, and is very good for capturing the general attitude towards a given brand. But the text-to-speech `API` we are using is quite crude. This means that in many of our test cases it gave us illogical transcriptions of the given vocal input. When this happens our program will also give very bad results, especially when the brand name itself is not correctly interpreted from the recording.

The least we can say is that we are very happy with the results. We hope that our client will be as happy with our experimental results.

1 Contributions

1.1 Jens

Jens wrote the basic interface code and the code communicating with/parsing replies from each of the API servers. He also carried out the experiments and wrote the Results section of the report.

1.2 Fredrick

Fredrick started out by researching text-to-speech analysis tools. He wrote the Background and Method sections of the report.

1.3 Erik

Erik wrote the code for inputting brands and searching for mentions of them in the text, as well as that for recording and converting file formats and presenting the results in the table. In the report he wrote the section about previous work in text-to-speech and the discussion section.

1.4 Petter

Petter did much on the report, mostly LaTeXthings to make this document look acceptable. But when it comes to writing he wrote the Abstract, Sentiment Analysis section and small parts of the Text-to-Speech section. Also, he prepared all the posters for the poster session. In the beginning he tried to find audio files (with brand mentions in them) to analyze. He also recorded some of his own.

2 Background

Today, we have more ways than ever of sharing information thanks to the Internet and social media. We can share our thoughts, feelings, and experiences with just a few button strokes. In addition, consumers today can reach a wide audience when they express their feelings about products and brands. This data on consumer preferences is of great value to companies. Companies can mine the data to extract trends and opinions from their clients. And yet, there is an oasis of spoken data relating to this area that remains unexploited.

At the same time, technology is moving forward on multiple levels. Portable devices are capable of recording better quality audio than they were ten years ago. Smartphones are also far more ubiquitous than they were at the time. Furthermore, new `APIs` are emerging that make it easy to process audio recordings. These technological advancements herald in opportunities that are of great value to companies. The possibility to effectively process live or recorded audio to elicit consumer thoughts and preferences may be just around the corner. Empowered with this information, companies will be able to position their product lines accordingly.

And yet, there are some unresolved issues. Data may be difficult to process for several reasons. One reason is the sheer amount of it. Another aspect of data that may make it difficult to process is its format. The spoken word is far more difficult for a computer to process than a written document. To capitalize on this treasure trove of information, we need to develop effective ways of deriving details from natural speech and processing them.

A large part of our media is still partly or fully in the form of sound, such as that of web radio, and television. It is hard to analyze this data, as the format is often not native to our computers.

With this is mind, our goal is to make an experimental program that can transcribe speech and analyze it, looking for certain trends. It may be used to recognize brand mentions in an audio stream, and then analyze the context to find an opinion of the brand. A complete solution could be very interesting for companies that care to know how their brand is doing “out there” in the world.

3 Related Work

3.1 Speech-to-Text

Speech-to-text, or more generally, *Computer Speech Recognition*, is a common problem in computer science where the computer has to translate spoken words into text.

The first real attempt at speech recognition took place during the 50’s. The system could only handle a small subset of speech, namely digit recognition. In 1952, researchers at Bell Laboratories built a speech recognition system that only could recognize uttered digits by a single speaker in a quiet room. The technology of the system was based on the theory of the acoustics of speech and the machine was triggered by the sound of vowels in the different digits.

Similar work was made by researchers at RCA Laboratories and at the MIT Lincoln Lab around the same period. The research teams at these two institutions built speech recognition systems that could classify vowels that a test person spoke.

In the 60's, there were several Japanese teams that made progress in speech recognition, especially two professors named Suzuki and Nakata at Kyoto University. Even though they still only could recognize single digits or vowels, these could now be recognized in continuous speech. The earlier systems assumed each utterance of the test speaker contained the complete digit or vowel and thus did not need to be actively segmented.

During the 70's work began on the first speech recognition system that could handle spoken words and sentences. A program named "Harpy" was made at Carnegie Mellon University and could recognize speech containing 1011 different words. It had a moderate performance. Harpy used a graph search algorithm to obtain its results.

A new technology based on the n-gram language model was introduced in the beginning of the 80's and has been very important to speech recognition systems ever since. One obvious use for a speech recognizer is in an automatic typewriter: a user speaks words and sentences to be written to a document. Since there is an underlying structure and grammar in all forms of language, not all letter orderings and word orderings are equally probable. In fact, some orderings are even wrong, i.e the words are misspelled or sentences are grammatically incorrect. The n-gram model looks at a sequence of words and tells how probable it is. That will help when doing the word recognition; we will get a prior for each word and thus use a `M.A.P` estimation for what the next word ought to be.

These typewriters were meant for use by individuals; thus they allowed for individual training of the system. This was not the case for another application of speech recognition, namely routing phone calls. Usually when calling support centers there is a need for routing phone calls to the correct expert or department. This requires a large workforce of operators, and could, if automated, save the companies a lot of money. These systems were to be used on demand by millions of people; thus they needed to generalize very well to different voices and regional accents. The utilization of this technology started in the early 90's and today such systems are annually routing billions of phone calls around the world.

A technology that was developed and became more prevalent during mid 80's was the hidden Markov model (`HMM`). It quickly became the best and most preferred method for speech recognition. In a hidden Markov model, state and observation are key concepts. States are what the actual quantity being estimated is equal to. The obvious choice is words, but there are modified models, and other quantities could be chosen to represent the states. The `HMM` models the state transitions between the words based on uncertain measurements and the probability of a new word given the previous word. The state transitions have certain probabilities and some transitions are more probable than others (that might result in bad grammar or weird sentences). The measurements are the speech itself, using sound waves captured from the microphone[1].

Further on more modern machine learning techniques such as support vector machines and artificial neural networks have been used to get promising results within the field.

Today the use of speech-to-text, or speech recognition, can be found in many different fields and applications. It is extensively used in industry and by customers. Hands-free computing, writing emails just by talking to the computer, voice recognition on the cellphone to determine whom to call or what music to listen to or perhaps new intelligent assistants such as Apple iPhone's Siri are helping people to do daily tasks with speech recognition. Google uses speech recognition similar to Siri on their Android phones, but also as text input to their search engine together with Google Chrome. This is the service we are using in our program.

3.2 Sentiment Analysis

The *Sentiment Analysis* or *Opinion Mining* problem is: Given any snippet of text, find the general attitude of the text regarding a topic. As a very simple example, the text "I hate Mondays" has a very negative attitude, while the attitude of "I would not prefer Mondays" is more neutral.

Some early work in this area consisted of sampling reviews of a movie or restaurant and analyzing the polarity of the samples. Early work on this has been attributed to Turney[3] and Pang[4].

Being able to get the general opinion or attitude from a text can be very useful for companies that care about their reputation. Thus, for example, finding a lot of complains about something could give great automatic feedback.

With the rapid rise of social media platforms such as *Twitter* and *Facebook*, the general interest in Sentiment Analysis has skyrocketed. Businesses are looking for methods of automating the mining of data from these platforms, ways to elicit public opinion, and in case of negative feedback, a means to immediately attend to the issue.

4 Method

Our solution to this problem consists of three steps, which are later combined into a single program.

1. Our first step is to work convert speech-to-text. For this we use Google's speech `API`.

2. Next, we process the converted text to capture mentions of the brands that the data miner is interested in.

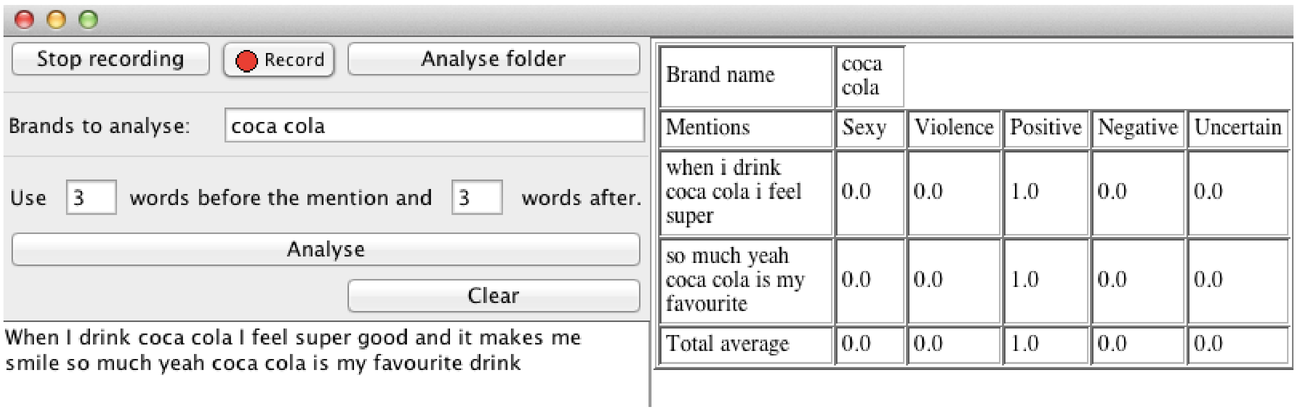
3. Finally, we process the expressions related to these brands to detect sentiments, using Gavagai's `API`.

Google's speech `API` is used widely across its product range, from web search to translate to Android voice commands. Provided with an audio file in `FLAC` format, the `API` returns a `JSON` object representing the transcribed audio. The `JSON` object can then be parsed to retrieve the text string that corresponds to the spoken message. Google's speech `API` allows for a number of different configuration settings, including language and audio frequency of the supplied file. However, it only allows for file sizes up to 256 kB, which means that larger recordings must be split up into smaller parts and submitted individually.

In our project, audio file input is extended by letting the user of the program record an audio snippet in real time and submit it for processing.

In the second step, mentions of brands are elicited from the transcribed audio. In our program, the user supplies the name of a brand that he or she wants to watch for. The user also chooses how many words before and after the brand name should be included in the analysis. Thus, there is a precision-vs-recall trade-off here. Choosing a large number of words to include in the processing increase the scope of the search, but might lower the precision as unrelated adjectives may be grouped with the brand mention.

In the third step, we provide cardinal scores of the audio according to a number of criteria. Gavagai's `API` lets us evaluate a text string based on a number of parameters. We can look for the sentiments described as positive, negative, sexy, violent, and uncertain. The program then presents the user with the processed data tabulated as shown in the figure.



Brands to analyse: coca cola

Use 3 words before the mention and 3 words after.

When I drink coca cola I feel super good and it makes me smile so much yeah coca cola is my favourite drink

Brand name	coca cola				
Mentions	Sexy	Violence	Positive	Negative	Uncertain
when i drink coca cola i feel super	0.0	0.0	1.0	0.0	0.0
so much yeah coca cola is my favourite	0.0	0.0	1.0	0.0	0.0
Total average	0.0	0.0	1.0	0.0	0.0

Our solution to this problem consists not of trying to reinvent the wheel, but of employing different solutions that are more-or-less complete throughout the different stages, and then stitching them together.

5 Results

The program was tested on a few different scenarios, to measure its performance under different conditions, as well as testing the limits of the frameworks used in it. The tests were done in a four-step process:

- Input text into the program and run a sentiment analysis for brands mentioned in it.
- Record one of the authors reading the text aloud, and input it into the program.
- Compare the sentiment scores of the text and the audio file.
- Compare the sentiment scores to the content of the text.

5.1 Big files

The test text used for this test was a long blog post detailing the author's love for the Coca-Cola soft drink. The sentiment scores for the text itself indicated that the text was both positive and sexy: Sexy: 0.42857142857142855 Violent: 0.14285714285714285 Positive: 1.2857142857142858 Negative: 0.07142857142857142

Unfortunately, due to limitations on big files, the output from he speech-to-text `API` was mostly unintelligible, and very little of the original text survived the conversion. Thus, the results were unusable for any sane comparison. This finding limited our testing to small files, and hindered testing on podcasts, live streams, and other interesting audio sources.

5.2 Mixed brands

In this test, short texts which mention multiple brands with different sentiments were used. The program was configured to use a small scope when searching for sentiments, 3 words behind and 3 words after the brand name. The audio file input was a failure, since most of the time brand names weren't picked up by the speech-to-text `API`, so no sentiment analysis could be done on these texts. Even if the brand names were inserted, differing sentiments in the same text meant that the sentiment analysis often gave bad results. An example is the sentence "I love Coke, but I hate my Ford. Although, Coke is really nice to drink while driving, so I guess the Ford isn't completely worthless." The score of Coke here is Sexy: 1.5, Positive: 1.5, Negative:1.0 and the score for Ford is Negative: 0.5

When read aloud and input to the program, this sentence became "I love I need so who is he nice to drink while driving so I guess the board in completely worthless".

Aside from seeming to have nothing to do with either Ford or Coca-Cola, the interpreted sentence has a general sentiment score of Sexy: 2.0, Positive: 2.0, Negative:2.0 Three other, similar text snippets were recorded and tested, and produced similar results not possible to compare with the original text snippets.

5.3 Small snippets, one brand, one sentiment

The last test was made to ascertain that our program worked at all as intended, as the examples are idealized and might not be encountered in a real situation. They do, however, overcome the limitations set on our experimental program by the outside services used to power it, and as such serve as an adequate proof of concept. The snippets used are small; they contain only one brand that is mentioned multiple times (Ford), and the whole snippet only contains one sentiment. The scope used was ten words before the brand mention, and twenty words after.

The results are presented below. First are the original text and its sentiment score, then the processed recording of it being read aloud and the associated sentiment score.

"I love Ford since they make the best cars I have one myself and it is not only very efficient but also quiet and has a lovely design My friend tipped me off about it and I've been happy with the car ever since."

Score: Sexy: 2.0 Positive: 4.0

"I love Ford they make the best cars I have one myself only there any fish in it also quite lovely night my friend to pick me up about it and I've been happy with the car."

Score: Sexy: 2.0, Positive: 4.0

"My friend just bought a Ford, and I can't say anything about it really. It's a pretty standard car, and I'm unsure about what made him choose it. He's kind of average, so maybe he likes a car that's as mundane as himself."

Score: Positive: 1.0, Negative: 1.0, Uncertain: 2.0

"my friend just bored and I can send about it it's a pretty standard and I'm unsure about with meetings so maybe like the car and a Nazi."

Score: Positive: 1.0, Negative: 1.0, Unsure: 2.0

Note: The sentiment analysis for this snippet was run on the whole text since the word "Ford" was not picked up, so this snippet would probably not be considered in a larger analysis.

"I can't believe that Ford still exists as a company. They make awful cars, and have a terrible history as one of the worst environment-crooks among the big auto companies. Don't buy Fords, people!"

Score: Negative: 2.0

"I can't believe the Ford still exist the company they make of a car and have a tear as one of the work environment cooks big auto companies don't buy Ford Depot"

No sentiments were picked up by the sentiment analysis

"Some guy just trashed my Ford and beat me up. He somehow punched right through my windshield, threatened by pull my car tires off and then ran away screaming. This day could not get any worse."

Score: Violence: 3.0, Uncertain: 1.0

"oMG I just passed by Ford beat me up he somehow right to my wii ready to pull my car tires off and then run away screen this day could not get any books"

Score: Violence: 1.0, Uncertain: 1.0

6 Discussion

The scope of this project was obviously too small to design our own speech recognition algorithm or sentiment analysis algorithm. We had to rely on third-party services such as the speech sentiment `API` from Gavagai and Google's unofficial speech recognition web service. Even though a large international company like Google uses cutting-edge technology for their speech recognition, we had major problems with the recognition of some words, even when the speaker was alone in a quiet room and spoke at normal speed. Recognizing the brand names themselves proved to be fairly hard. Sometimes only one out of several mentions was recognized, making the end results worse.

When we first began with this project we tried to use CMU Sphinx, a speech recognition system developed at Carnegie Mellon University, now open source. However, the setup showed to be very cumbersome and we simply could not manage to make it work. Given the short time frame we turned to Google's unofficial web service. We figured we would still get as good a result as possible since Google uses the same speech-to-text service in its products.

The text-to-speech function did however do well when using short recordings or sound files as opposed to long recordings, which did comparatively worse. The reason for this was that we had to break up the sound files into segments of 256 kB and classify them one by one. This had to be done due to Google's own restrictions on large files. If a sentence was broken up just in the middle it sometimes resulted in grammatically incorrect phrases. Since the speech-to-text algorithms are based on grammar this further worsened the speech recognition results.

As for the sentiment analysis it preformed as expected and did fairly well if the text was typed in or pasted by hand. Certainly it performed better the more data that was available, so larger texts produced more precise results.

The program allowed the input of several brand names and could perform Sentiment Analysis for them at the same time on the corpus. When speaking of several brands at the same time in a recording, the algorithm produced very bad results. Firstly the speech recognition is worse when mentioning more special words that cannot be put in a grammatically correct context, and secondly the Sentiment Analysis does not seem to care that much about grammar or order of the words in the input. When doing multiple-brand Sentiment Analysis where the brands were mentioned close to each other they sometimes seemed to be mixed up.

Our program did produce reasonable results when we just mentioned one brand in a short sound file with the same sentiment towards the brand all throughout the recording.

References

[1] B.H. Juang & Lawrence R. Rabiner, *Automatic Speech Recognition A Brief History of the Technology Development*. Georgia Institute of Technology, Atlanta Rutgers University and the University of California, Santa Barbara

[2] Rabiner, Lawrence R. *A tutorial on hidden Markov models and selected applications in speech recognition.* Proceedings of the IEEE 77.2, pp. 257-286 1989.

[3] Peter Turney, *Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews*, Proceedings of the Association for Computational Linguistics, pp. 417-424 2002.

[4] Bo Pang; Lillian Lee and Shivakumar Vaithyanathan, *Thumbs up? Sentiment Classification using Machine Learning Techniques*, Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). pp. 79-86, 2002.

[5] Shanahan, James G., Yan Qu, and Janyce Wiebe, eds. *Computing attitude and affect in text: theory and applications.* Springer, Vol. 20. 2006.