Thesis Preparation Report

Kristoffer Videbæk Kunkel and Nina Holm-Jensen

11/01-2016

Supervisor: Rico Jacob

Problem Description

Discourse analysis is a wide and varied field, spanning the subject of language and communication in terms of tone, meaning, genres and subcultures, power, cognition, etc.. Many of these are qualitative or social sciences, not yet implemented into computer science methods.

One topic in discourse analysis, however, which has received increasing attention in the world of text analysis and information retrieval is that of the *relationship between terms*. This is used keenly in semantic analysis where the goal is to characterize the tone of a statement (or the use of a term) based on the surrounding text, and it is extensively used in information retrieval's multiple-term queries.

In this project, we understand discourse as "not a unit of semiotic signs, but an abstract construct that allows the semiotic signs to assign meaning, and so communicate specific, repeatable communications to, between, and among objects, subjects, and statements." (Wikipedia: Discourse) Namely, the discourse is, abstractly speaking, what causes words to have a meaning. The difference in discourse determines, for instance, that upon hearing the word "dog", a Dane thinks of pets, and a Chinese person thinks of dinner.

This meaning is partly derived from the relationship between semiotic signs (a semiotic sign can be understood, in the context of this report, as a word). Determining the frequency and proximity of words in relation to each other (such as "dog, recipe" or "dog, pet") is one way to examine the discourse of a given culture or context.

It is also the way this project approaches the subject. We will create a Discourse Machine which can examine the specific discourse of a body of text (namely, a newspaper's articles) by finding the most meaningful terms related to an input term. This could allow e.g. a researcher or a journalist to examine the difference between and the bias of available news outlets.

Problem Statement

We will create a search engine that, upon receiving an input term, returns the most meaningful words associated with the term for a given corpus of documents.

The significant challenges in this project will be:

Defining meaningfulness, related to the domain of newspaper articles or in the general case.
Ensuring scalability and performance for large numbers of documents.
Building a useful prototype.

Initial Prototype

We have developed an initial, simple prototype as proof of concept. The prototype reads any number of corpuses containing any number of documents.

To retrieve the documents, we built a web crawler to get news articles from four major news outlets. The news outlets served as individual corpuses, and the articles were - to avoid copyright issues - stored as a list of word counts. A more nuanced prototype would need more than that, but for the initial prototype, it was enough.

We then built a program which, upon receiving an input term, searched for the documents containing the words. In this subset, the average most important terms (which were not the input term) are calculated and extracted. This calculation is done for every corpus, leading to the X most important terms associated with the input term in every corpus.

In this initial prototype, the most important term is calculated using the Term Frequency Inverse Document Frequency (TF-IDF), a common measure for importance in information retrieval.

TF-IDF can be expressed as:

$$\frac{\text{Number of times term t appears in a document}}{\text{Total number of terms in the document}} \times \log\left(\frac{\text{Total number of documents}}{\text{Number of documents with term t in it}}\right)$$

This proof-of-concept prototype neatly illustrates the type of program we will create. However, it is neither scalable nor is the result terribly interesting. These are the things we will explore in the actual thesis project.

Non-trivial problems

We have zeroed in on our problem and made it tangible with our prototype which begs the question; what now? So far, we have identified five non-trivial challenges, all demanding research and creativity to solve:

Choice of data
Stemming and itemization
Defining meaningfulness
Scalability and performance
Testing

All of them are strongly interconnected. How sophisticated can we make our algorithm while still maintaining an acceptable performance? How do we test the validity of our results if our goal is to find something intuitively meaningful? In the following, we will touch upon these subjects, what we know, and what we hope to find out.

Data

In terms of computer science, the interesting parts of this project - defining meaningfulness and ensuring performance - are possible with any dataset. In terms of personal motivation, we are very interesting in doing an analysis of Danish newspapers. We have tried reaching out to both Infomedia and several newspapers directly to avoid the copyright issues in obtaining the data on our own through e.g. webcrawling. Data in this case being newspaper articles. We hope that our prototype and increased understanding of the problem will help us establish a deal with one or more newspapers. Since scalability is an important goal for us (and since we might need a certain data size to properly extract meaningful results), we are currently attempting to find a suitably large dataset.

While the topic of Danish newspapers is important to us, it is not crucial to the thesis. Due to the initial difficulty in obtaining the data we wanted, we have started lining up alternatives. Depending on how much we decide is sufficient, we could go for one of Reuters widely used corpuses. Reuters even offers a Danish corpus of their newswires from 20th of August 1996 to 19th of August 1997, containing over 487000 documents (http://trec.nist.gov/data/reuters/reuters.html). We could also

choose to go for English corpuses, which opens up the possibilities considerably. At present, we have sent in the necessary documents to Reuters and expect to hear back within seven days. We chose to do this early, so we have a safe backup set of data.
We also have a lead on a corpus of Danish news articles from the 1980s, which is used in research by the Center for Language Technology at the University of Copenhagen (CTS-KU). More on this later.

Our main challenge for data is, at present, to get our hands on it. We insist on finding text written in a journalistic context, since our interest lies in discovering bias in the public discourse, over which journalism has a significant influence.

Analysis

The core of our project is of course the design of the program used to process our data. We see the choices made at every step having an impact on our final result. We have already explained the basics of the TF-IDF algorithm used in our prototype. This solution potentially contains several variations of varying sophistication, especially in terms of normalization, but there is so much more we could do.
We have also talked extensively about stemming, the technique of reducing words to their base form. We have a few 'off the shelf' options lined up for the Danish languages but have not yet tested any. Alternative options are primitive stemming, using regexes, chopping off common Danish suffixes (er, et, ene, etc.), but we are hesitant to use too much time on this if an acceptable solution is present. This touches on a central point at this stage of the project, which is how much we do not know in the field of natural language processing of the Danish language. This brings us back to CTS at KU, who have undoubtedly faced some of these challenges beforehand and might be able to guide us in the right direction on several of the key issues, so we can skip making time-wasting, rudimentary discoveries. CTS is an ideal ally for us to make (http://cst.ku.dk/english/). We will start by going through research of theirs, and might contact them if our problems align.

Performance

Performance and scalability are vital to us. Getting a highly scalable and a meaningfully nuanced solution will be a constant trade-off, yet higher scalability will also afford greater nuance in the analysis. It is thus imperative that we spend time and effort optimizing the solution. The exact

capacity of the machine will naturally depend on the size of the final dataset and the sophistication of our analysis thereof, and we have already had several discussions with our supervisor on how to keep our eyes on this side of the ball. We will have to answer questions of how to efficiently structure the data, how much pre-processing of intermediate computed values is possible, and how to scale/update the data in what we hope to be an evolving machine, churning through the public discourse. Fortunately, this falls well within our supervisor's field.

Testing

Testing in our contexts holds a certain duality to it. First we have the performance tests which are absolutely necessary to make any meaningful judgements on a program like the one we have proposed. We will draw extensively on education received here at ITU to design and conclude on performance tests. The second testing category depends on how we define meaningfulness. We must make precise and enforceable definitions of what "meaningful" means in the context of this solution. These definitions will undoubtedly depend on the nature of our data. This also poses the interesting question of what is to blame if the program outputs unexpected or undesired results; the algorithm's design or our preconceptions? Our background in Communication (bachelor degree) will help us discuss and refine these tests in a way that combines human intuition (what is interesting) and computer logic (what can be mathematically described).

Methodology

Due to the refinement-based nature of this project, and the fact that we already have a functioning prototype, an agile approach seems obvious. We will cover the project in a number of iterations, each covering part of the requirements, and each resulting in a functional product. The discourse machine will thus go from a rough word counter to a scalable, sophisticated machine.

Initially, we will conduct our work on only one dataset, introducing comparable analysis in a later iteration.

The topics on which we need to focus in separate iterations are:

> *Data collection and preprocessing*. This includes obtaining and familiarizing ourselves with the data, deciding on initial data structures and storage. In this iteration, we must also decide on tools and requirements for later analysis, such as data size, potential need for a virtual machine for analysis, etc.
>
> *Define meaningfulness*. This includes distilling the discursive understanding of "meaningful" to a mathematical construct, designing tests, initial coding of the logic.
>
> *Set up (scalable) architecture and environment*. This includes setting up a big data environment (maybe going from test data to full dataset) and designing tests for the last iteration.
>
> *Introduce comparison of different corpuses*. This includes obtaining and cleaning the data, and implementing any cross-corpus analysis.
>
> *Test for scalability and meaningfulness*. This includes ensuring that everything runs smoothly, and determining the relationship between data size and meaningfulness.

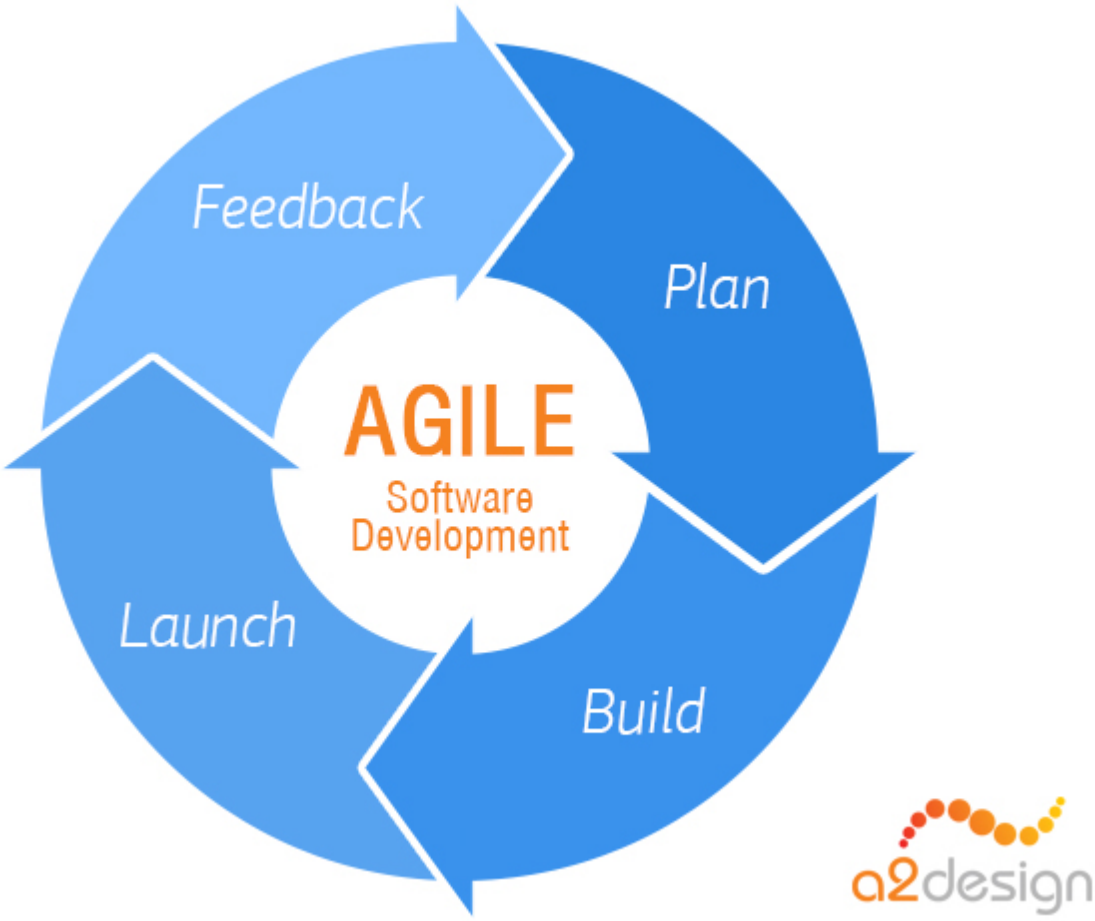
Due to the small group size, a full scrum process is not necessary. We will, however, break down each iteration into four, classically agile phases:

Plan. Research, solidify theoretical background, define specifications.

Build. Implement theory into solution.

Launch. Test solution, supervisor meeting.

Feedback. Implement changes, fix bugs.



In the planning and building phases, our supervisor takes on the role of technical counselor, introducing us to the tools necessary to solve the task, and helping us create precise time estimates. In the launch phase, our supervisor takes on the role of client.

After every iteration, we deliver:

An iteratively improved, functional discourse machine

A report chapter describing the theory used and learning goals achieved, to ensure that our code stays thoroughly grounded in theory.

Schedule

We hope to schedule a set weekly supervisor meeting to structure our work and to minimize any misunderstandings or misguidance. This has not yet been discussed nor confirmed with our supervisor. Neither has the schedule below, which might thus be revised after the next supervisor meeting.

	Official deadlines	**Iteration**	**Description**
Week 5 - 01/02	Official start of thesis	Data iteration	
Week 6 - 08/02		Data iteration	
Week 7 - 15/02	Thesis agreement and plan, February 15	Data iteration	Collect and clean data, familiarize ourselves with data. Collect readings and theory necessary for further work.
Week 8 - 22/02		Meaningfulness iteration	
Week 9 - 29/02		Meaningfulness iteration	Define meaningfulness in terms of the discourse machine. Incorporate the logic into the primitive machine.
Week 10 - 07/03		Meaningfulness iteration	
Week 11 - 14/03		Scalability iteration	
Week 12 - 21/03		Scalability iteration	
Week 13 - 28/03		Scalability iteration	Set up scalable environment.
Week 14 - 04/04		Comparison iteration	
Week 15 - 11/04		Comparison iteration	Gather additional datasets, set up comparison.

Week 16 - 18/04		Comparison iteration	
Week 17 - 25/04		Test iteration	
Week 18 - 02/05		Test iteration	
Week 19 - 09/05		Test iteration	Test meaningfulness and scalability of the solution.
Week 20 - 16/05		Write report	Put together all previous written components and give the report its final structure.
Week 21 - 23/05		Catch-up week	Tie up loose ends, refine the report.
Week 22 - 30/05	Thesis deadline, June 1	Deliver	

The supervisor will be absent for the entirety of February, meaning that supervision will happen digitally in that time.

Both members of the thesis group are working jobs 1-2 days of every week. Both can be flexibly coordinated. We will also take a few days off at the end of March, to provide an extended weekend vacation. No longer breaks are planned during the project period.