Aalto University
School of Science
Master's Programme in ICT Innovation

Hendrik Heuer

Semantic and stylistic text analysis and text summary evaluation

Master's Thesis

Stockholm, July 20, 2015

Supervisors: Prof. Jussi Karlgren, KTH Royal Institute of Technology
Prof. Samuel Kaski, Aalto University

Instructor: Dr. Jorma Laaksonen, Aalto University

Aalto University
School of Science

Aalto University School of Science Master's Programme in ICT Innovation	ABSTRACT OF THE MASTER'S THESIS

Author: Hendrik Heuer

Title: Semantic and stylistic text analysis and text summary evaluation

Number of pages: 46 | Date: 20.07.2015 | Language: English

Professorship: Computer Science | Code: T-61

Supervisor: Prof. Jussi Karlgren and Prof. Samuel Kaski

Instructor: Dr. Jorma Laaksonen

Abstract:

The main contribution of this Master's thesis is a novel way of doing text comparison using word vector representations (word2vec) and dimensionality reduction (t-SNE). This yields a bird's-eye view of different text sources, including text summaries and their source material, and enables users to explore a text source like a geographical map.

The main goal of the thesis was to support the quality control and quality assurance efforts of a company. This goal was operationalized and subdivided into several modules. In this thesis, the Topic and Topic Comparison modules are described.

For each module, the state of the art in natural language processing and machine learning research was investigated and applied. The implementation section of this thesis discusses what each module does, how it relates to theory, how the module is implemented, the motivation for the chosen approach and self-criticism.

The thesis also describes how to derive a text quality gold standard using machine learning.

Keywords: Text Analysis, Machine Learning, Distributional Semantics, Word Representations, Dimensionality Reduction, word2vec, t-SNE, Sentiment Analysis, Topic Modelling

Contents

1 Introduction

This thesis focuses on semantic and stylistic text analysis and text summary evaluation in theory and in practice. For this thesis, the use case and requirements of a company are operationalized, the state-of-the-art in research in regard to these requirements is investigated and a web application that makes this research usable is implemented. Unfortunately, this public thesis does lack a lot of information due to an NDA.

The thesis introduces a novel way of comparing text sources. This approach uses word vector representations and dimensionality reduction to visualize and compare the topics in text summaries and their source material using an interface that is similar to a geographical map. The thesis also describes how to derive a text quality gold standard using machine learning.

1.1 Requirement analysis

The main goal of the thesis was to aid the quality assurance and quality control efforts of a company. The research goal of this thesis was to investigate the state of the art of semantic and stylistic text analysis. An important aspect was to ensure that a text summary accurately reflects the text it summarizes. For this, a way to ensure that a summary covers all or an appropriate subset of the stories and topics of the source material was needed. The tool developed for this was made available under GNU General Public License 3 [20].

During the requirement analysis phase of the project, a literature review was compiled. This informed the discussions with the company. The discussions with the company also helped to understand the company's use case and their intentions. Based on this, a variety of different modules was proposed, implemented and tested. During this phase, experiments and small ad-hoc user tests were conducted to understand the possibilities and limitations of the available tools and approaches.

1.2 Operationalization

The main problem this thesis is addressing is comparing two text sources to each other. These two text sources are not independent of each other: A text is compared to a summary of that text. Conceptually, the goal is to find the most salient features in the source material. The following section gives a high-level overview on how this thesis addresses these requirements.

1. *Readability and reading time*

   Readability scores can be used as a proxy to assess and evaluate the complexity of terminology and sentences. Readability tests like the Flesch–Kincaid were developed to quantify how difficult a text is to understand. For this, word length and sentence length are the most important indicators.

   The reading time is computed based on the number of words in a text and the average reading speed of an adult.

2. *Stories*

   Stories can be described as narratives on certain themes which relate to many different topics at once. An important outcome of the requirements analysis was to understand the intention to model stories, which was to visualize the number of stories in a text and to help users find stories. Therefore, an approach that relies on Latent Dirichlet Allocation (LDA) was deemed as a good operationalization of this concept.

3. *Topics*

   To analyse the topics in a text, it is important to be able to automatically and flawlessly detect them. For this, a topic module that enables the user to explore all the concepts in a text was designed and developed. Even in regards to human judgment, it is hard to define what the main topic of a sentence is and how to automatically detect it.

1.3 Organization of the thesis

In this *1. Introduction*, the use case of the company is described and operationalized. The *2. Background* chapter discusses the theoretical and conceptual background and the corresponding literature. In *3. Distributional semantics and word representations*, a detailed overview of distributional semantics and word representations is provided as this is fundamental to the implementation of the topic and topic comparison approach described in this thesis. The *4. Implementation* chapter discusses the modules. For each module, a description of what it does, how it relates to the theory, how it is implemented, the motivation for the approach and self-criticism are provided. In the *5. Experiments and results* chapter, findings from the implementation of the modules are put into a larger context. The chapter also outlines how the findings could be evaluated. In *6. Conclusions*, the findings and its implementation are assessed. The last chapter, *7. Future Work*, discusses how the findings could be extended and improved in the future.

2 Background

2.1 Text summarization

The thesis topic is indirectly connected to the problem of text summarization. While the aim of this thesis is not to automatically summarize text, automatically generated summaries can inform and help evaluate the human-created summaries. In regard to text summarization, a variety of different approaches can be used, including, but not limited to: Hidden Markov Models [9], Graph-based approaches [11], and probability distribution based approaches [33]. In the probability distribution based approach, the similarity between a source text and its summary is quantified and a similarity score, which replicates human assessment, is computed [33].

For this thesis, probability distribution based approaches are especially interesting, because of their explanatory power that might inform the process of how the human-created summaries are created. Louis and Nenkova showed that good summaries can be characterized by a low divergence between the probability distributions of words in a certain text and the distribution of words in a certain summary. Bad summaries can be characterized by a high divergence between the probability distributions. They used the following three measures to compute the divergence: Kullback-Leibler (KL), Jensen-Shannon (JS), and cosine similarity [33]. Their work is available as a Java open-source tool called SIMetrix (Summary Input similarity Metrics) [33].

2.2 Sentiment analysis

Sentiment analysis aims to identify the viewpoints underlying a text span [44]. The task can include polarity classification as well as assigning arbitrary labels. With polarity classification, a document is either negative or positive (thumbs up or thumbs down). The arbitrary labels can include *positivity*, *negativity*, *fear*, *hate*, *love*, *skepticism*, *violence*, and *desire*. Agarwal et al. indicate that sentiment analysis started as a document-level classification task [56, 44], but was extended to the sentence level [21, 27] as well as the phrase level [60, 1]

Yi et al. showed that it is possible to associate the opinion to a topic using term co-occurrence in the same context, even though it is hard to associate a sentiment to a specific topic [63]. Misattributions are another limitation or problem that can occur with simplistic models. A too naïve approach is to look for positive or negative sentiment on a word level, e.g. by using a lookup table. This fails to capture the sentiment of phrases and entire sentences. Socher et al. illustrate this with the sentence: "This film doesn't care about cleverness, wit or any other kind of intelligent humor". On a word level, this sentence includes a variety of positive and very positive words such as *care*, *cleverness*, *intelligence* and *humor*. However, the *does not* negates the sentiment of the sentence. Therefore, the sentence as a whole is negative. Most approaches in sentiment analysis use bag-of-words representations [44]. With a naïve bag-of-words approach that only regards the sentiment associated with single words or small n-grams ($n < 13$), the sentence in the example would be

impossible to classify correctly.

According to Socher, bag-of-words classifiers can work well for longer documents by relying on a few words with strong sentiment like *awesome* or *exhilarating*, but this fails to exceed classification performance above 80% [54].

2.2.1 Recursive Neural Networks

Socher et al. explored the usage of a Recursive Neural Model (RNM) that computes a compositional vector representations for phrases of variable length and syntactic type [54]. Based on this, Socher et al. introduced the Recursive Neural Tensor Network (RNTN), which has a single, powerful composition function and which composes aggregate meaning from smaller constituents more accurately [54]. An RNTN takes input phrases of arbitrary length and represents a phrase through word vectors and a parse tree. The RNTN uses the same composition function to compute vectors for higher nodes in the parse tree [54].

For single sentence sentiment detection, the RNTN pushes the state of the art for positive/negative sentence classification by 5.4%. The RNTN receives 80.7% accuracy on the sentiment prediction across all phrases and captures negation of different sentiments [54]. This approach is available as part of the Stanford CoreNLP Java tool [34].

2.2.2 Multidimensional tonality analysis

Karlgren et al. discuss a consumer attitude scenario, where *recommend*, *endorse*, *surprise*, *disappoint*, *satisfy* or *delight* are more salient examples of attitudes than a polarity such as *positive* or *negative* [25].

Karlgren et al. argue in favour of a knowledge model with an application in mind instead of a focus on the positive-negative dichotomy perpetuated by the available benchmarks, especially since they believe this effort is misguided. Regarding human performance at the task, they conclude that it might be impossible to do well at a positive-negative classification task and that it is likely to be of little application potential if optimised beyond its reasonable level of accuracy [25].

A knowledge representation for sentiment analysis of text for real world applications must be multi-polar and not restricted to positive and negative sentiment [25]. Their research led to Gavagai, a company from Stockholm, Sweden, which was started by researchers from the Swedish Institute of Computer Science (SICS). Gavagai provides a sentiment analysis API which offers multidimensional tonality analysis and computes scores for *positivity*, *negativity*, *fear*, *hate*, *love*, *skepticism*, *violence*, and *desire*.

2.3 Dimensionality reduction & visualization

For this thesis, PCA and t-SNE dimensionality reduction techniques were used to visualize high-dimensional word vectors. Both PCA and t-SNE can project high-dimensional data down to 2D or 3D, where the data can be easily visualized and interpreted by humans [45].

2.3.1 PCA

Principal Component Analysis (PCA) is a linear dimensionality reduction technique. The goal of PCA is to quantify the importance of each dimension of a data set for describing the variability of the data set. PCA aims to find a new vector basis, which best re-expresses a data set and which is a linear combination of the original vector basis [51]. The standard way of calculating PCA is doing a Singular Value Decomposition (SVD), a factorization of a matrix, to get the most significant singular vectors to project the data into a lower dimensional space [45]. SVD ranks the singular values (eigenvalues) in descending order. Principal components that come first contain a relatively larger amount of information than the principal components that come later. Only a few of these principal components are enough to describe the data set [45].

2.3.2 t-SNE

t-distributed Stochastic Neighbour Embedding (t-SNE) is a dimensionality reduction technique that retains the local structure of data and that helps to visualize large real-world datasets with limited computational demands [57]. Vectors that are similar in a high-dimensional vector space get represented by two– or three–dimensional vectors that are close to each other in the two– or three–dimensional vector space. Dissimilar high-dimensional vectors are distant in the two– or three–dimensional vector space. Meanwhile, the global structure of the data is revealed.

t-SNE achieves this by minimizing the Kullback-Leibler divergence between the joint probabilities of the high-dimensional data and the low-dimensional representation. The Kullback-Leibler divergence measures the dissimilarity ("distance") of two probability distributions by a discrete scalar and equals zero if they are the same [57].

2.3.3 Visualization

The Topic Comparison tool uses a variety of different JavaScript toolkits to visualize the data including D3.js and Google's Graph API. For most of these toolkits, data is exchanged using the JSON format.

2.4 Digital Humanities & Computational Social Science

A combination of techniques from computer science and social sciences opens up new possibilities for research. It enables the analysis of massive social networks and millions of books.

The emerging fields of digital humanities and computational social science leverage the capacity to collect and analyse data at a scale and may reveal patterns of individual and group behaviours [28]. Natural language processing and machine learning techniques have been used to conduct research on journalism. They made large-scale investigations possible like the analysis of 2.5 million articles from 498 different English-language news outlets [14]. Texts can be automatically annotated into topic areas. These topics can be compared in regards to readability, linguistic subjectivity and gender imbalances [14].

Jockers used computer science techniques to show that external factors such as author gender, author nationality, and date of publication affect the choice of literary themes in novels [22]. Using statistical methods, he identified and extracted hundreds of topics from a corpus of 3346 works of 19th-century British, Irish, and American fiction and used these topics as a measurable, data-driven proxy for literary themes [22]. The 500 topics used by Jockers for his book Macroanalysis can be downloaded [23]. For each topic, the 200 most related and probable words are published [23]. The Mallet implementation of Latent Dirichlet Allocation was used to compute themes based on word co-occurrence patterns [22].

2.5 Metacognition & learning theory

The book series *Head First – Brain-Friendly Guides* takes findings from metacognition and learning theory into account to communicate their content more effectively [43].

The "Head First learning principles" advocate making the content more visual. They cite that images are far more memorable and make learning much more effective, which according to O'Reilly can lead to a 89% improvement in recall and transfer studies [43]. They claim that students performed up to 40% better on post-learning tests if the content spoke directly to the reader in a first-person, conversational style [43]. Other studies support the notion that students learn better when words are in conversational style rather than formal style [35].

Other "Head First learning principles" recommend to include challenges, exercises, and thought-provoking questions, as well as activities that involve both sides of the brain and multiple senses. They also recommend touching the reader's emotions [35].

2.6 Human memory

Peter Gray defines memory as "an individual's entire mental store of information and the set of processes that allow the individual to recall and use that information when needed". Memory itself is not a single entity, but a whole range of phenomena that are classified as memory [17].

In regards to effective encoding strategies, Miller introduced the concept of chunking [39], where the goal is to reduce the number of separate items, both to decrease the number of items to be remembered and to increase of amount of information associated with each item.

Generally, memory refers to all effects of prior experience on subsequent behaviour and involves conscious and unconscious mental processes [17]. Therefore, a useful operationalization of how the human memory works is hard.

Ebbinghaus was a pioneer of memory research who experimentally investigated how the human memory works. Ebbinghaus' "Memory: A contribution to experimental psychology" was published in German in 1885 and in English in 1913 [10]. To study human memory, Ebbinghaus memorized series of nonsense syllables for over one year (1879-80), and then replicated the entire experiment again (1883-4) before publishing it.

Ebbinghaus' forgetting curve (Figure 1) describes an exponential loss of memory unless the information is reinforced [55]. The spacing effect describes that it is more effective to distribute trials over time than learning everything in a single session [62].

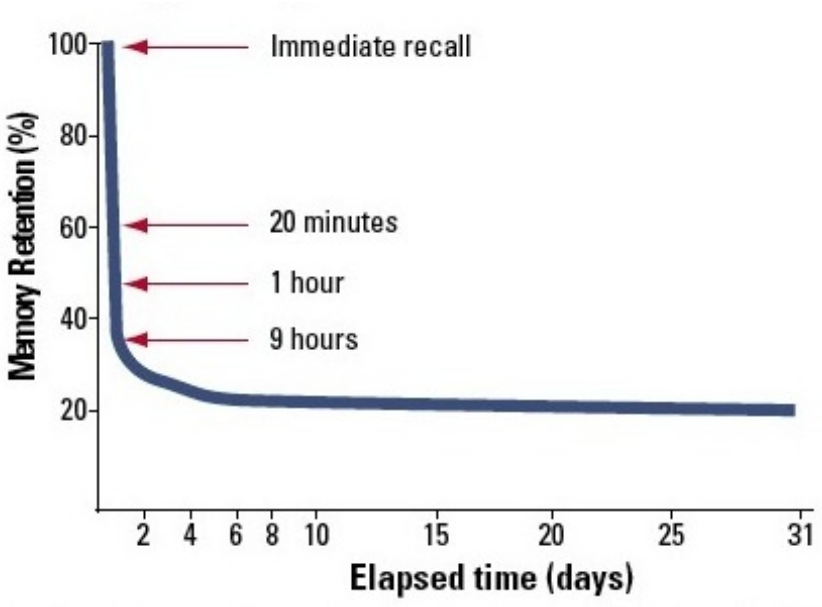


Figure 1: The forgetting curve according to Ebbinghaus. The graph shows Ebbinghaus' retention of nonsense syllables [55].

Wozniak notes that in the century since the publication of Ebbinghaus' monograph, surprisingly little has been learned about rote learning and retention that was not already known to Ebbinghaus [62].

3 Distributional semantics & word representations

The distributional hypothesis by Harris states that words with similar meaning occur in similar contexts [48]. This implies that the meaning of words can be inferred from its distribution across contexts. Bruni et al. showed that this claim has multiple theoretical roots in psychology, structuralist linguistics, lexicography and philosophy (e.g. in the late writings of Wittgenstein) [7, 13, 19, 41, 61]. As Firth famously said: "You shall know a word by the company it keeps!" [13].

The traditional approach to statistical modelling of language is based on counting frequencies of occurrences of short symbol sequences of length up to N and did not exploit distributed representations [30]. The general idea behind word space models is to use distributional statistics to generate high-dimensional vector spaces, where a word is represented by a context vector that encodes semantic similarity [48].

The goal of Distributional Semantics is to find a representation, e.g. a vector, that approximates the meaning of a word (see Figure 2) [7]. The Distributional Hypothesis states that terms with similar distributional properties have similar meaning [48]. This semantic similarity is computed by comparing distributional representations [49]. According to Schütze, vector similarity is the only information present in word spaces. This implies that semantically related words are close and semantically unrelated words are distant [50]. The representations are called distributed representations because the features are not mutually exclusive and because their configurations correspond to the variations seen in the observed data [30]. LeCun provides the example of a news story. When the task is to predict the next word in a news story, the learned word vectors for Tuesday and Wednesday will be very similar as they can be easily replaced by each other when used in a sentence. The same applies to the word vectors for Sweden and Norway [30].

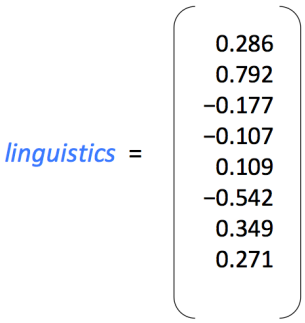


Figure 2: The word *linguistics* represented by a word vector [53].

There are a variety of computational models that implement the Distributional Hypothesis. The Distributional Semantic Model generally uses a co-occurrence matrix where the columns represent a term and the rows represent a context. For terms and contexts, the co-occurrence frequencies are counted. The rows are n-dimensional distributional vectors. Using this model, terms that occur in similar contexts get similar distributional vectors [49]. When word vectors are trained on

large collections of data, they provide good generalization and they can be used as features in many different NLP tasks [8].

3.1 Similarity encoding

An interesting observation about these vector spaces is that certain vectors can encode and represent high-level concepts. Mikolov et al. showed that the vector from man to women is very similar to the vector from uncle to aunt and from king to queen (Figure 3) [36, 38]. This vector could be described as the high-level concept of *femaleness* or *gender*.

Word vectors encode semantic meaning and capture many different degrees of similarity. In this vector space, linear algebra can be used to exploit the encoded dimensions of similarity. Using this, a computer system can complete tasks like the Scholastic Assessment Test (SAT) analogy quizzes, that measure relational similarity. An example for an SAT-style analogy task is the question: "man is to woman as king is to what?". The system can identify that king is equivalent to queen:

$$king - man + women = \textbf{queen} \tag{1}$$

It works for the *superlative*:

$$fastest - fast + slow = \textbf{slowest} \tag{2}$$

As well as the *past participle*:

$$woken - wake + be = \textbf{been} \tag{3}$$

It can infer the Finnish national sport from the German national sport.

$$football - Germany + Finland = \textbf{hockey} \tag{4}$$

Based on the last name of the current Prime Minister of the United Kingdom, it identifies the last name of the German *Bundeskanzlerin*:

$$Cameron - England + Germany = \textbf{Merkel} \tag{5}$$

The analogies can also be applied to the national dish of a country:

$$haggis - Scotland + Germany = \textbf{Currywurst} \tag{6}$$

Mikolov et al. showed how to solve this by doing a nearest neighbour search in the continuous space word representation [36]. In the vector space, the cosine distance between two vectors can be interpreted as how similar the represented words are to each other. If probed for the nearest neighbours of the word *Sweden*, the Topic Comparison tool outputs *Norway*, *Denmark* and *Finland* as the most similar results. Thus, the names of other Nordic countries are seen as the most similar to the word *Sweden*. Figure 4 shows the other results, which include a variety of European countries. These results are both semantically and structurally sound.

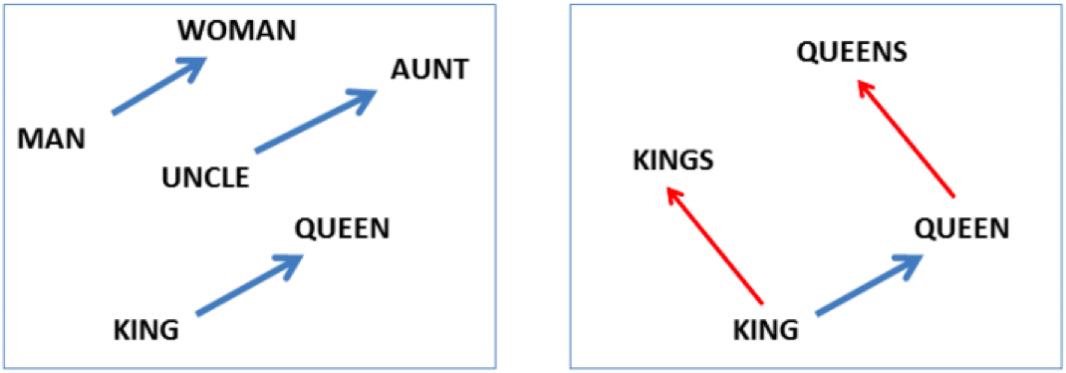


Figure 3: Semantic relationships encoded by word vector representations (Blue: Gender, red: Plural) [38, 37].

Interestingly, the adjective *swedish* is ranked below *Helsinki Vantaa*. *Helsinki Vantaa* is a city and municipality in the Helsinki Metropolitan Area where the main airport of Helsinki is located. Hence, the word representation captures the fact that *Helsinki Vantaa* is a place connected to *Sweden*, presumably as it appears in the context of people traveling from Sweden to Finland by airplane. Meanwhile *swedish* as an adjective is represented as structurally less similar to *Helsinki Vantaa*. This can be explained by the fact that the vectors capture semantic and syntactic similarities. While *Helsinki Vantaa* is a location like *Sweden*, *swedish* as an adjective is a property.

The similarity comparison also includes seemingly outliers like *Floorball Federation*. However, according to Wikipedia, floorball was developed in the 1970s in Sweden and is most popular where it has developed the longest [58]. Thus it is a noun that is closely connected to Sweden, as it represents what people associate with Sweden as a country and a word (as in: semiotic object and representamen).

All this exemplifies how the model is trained and connected to the context in which the word *Sweden* appeared in the training corpus. In conclusion, word2vec word vectors can capture many linguistic properties such as gender, tense, plurality and even semantic concepts such as *is capital city of*.

3.2 Word vector representations

One of the fundamental challenges of distributional semantics is computing word vectors that are suitable representations of words. There are various architectures that are used to compute such word vectors. In one tradition, word vectors are trained as part of a neural network language model with an input layer, a linear projection layer, a nonlinear hidden layer and an output layer with a softmax [3]. However, this basic model is computationally very expensive. According to Mikolov, the artificial neural networks can be thought of as a nonlinear projection of data [36]. A neural network function should ensure that semantically similar sentences are represented similarly.

Word	Cosine distance
norway	0.760124
denmark	0.715460
finland	0.620022
switzerland	0.588132
belgium	0.585835
netherlands	0.574631
iceland	0.562368
estonia	0.547621
slovenia	0.531408
floorball_federation	0.529570
luxembourg	0.529477
czech_republic	0.528778
slovakia	0.526340
romania	0.524281
kista	0.522488
helsinki_vantaa	0.519936
swedish	0.519901
balrog_ik	0.514556
portugal	0.502495
russia	0.500196
slovakia_slovenia	0.496051
ukraine	0.495712

Figure 4: Cosine similarity for the vector representing *Sweden*.

With neural networks, there are a variety of hyper-parameters involved, which have to be tweaked manually. However, according to Collobert, the choice of hyperparameters such as the number of hidden units has limited impact on the generalization performance [8]. The Topic Comparison tool and this thesis rely on a variety of open-source implementations of the approaches discussed and use the default parameters of the tools.

Generally, there is nothing that neural networks can do in Natural Language Processing that symbolic natural language processing techniques fail at, but neural networks outperform the traditional methods in competitions and gain better accuracy.

Ways of computing word vector representations include word2vec [36], GloVe [46], Dependency-based word embeddings [31] and Random Indexing [48]. The following sections give a short overview of each approach.

3.3 word2vec

word2vec is the name of a tool that implements two different ways of computing word representations: continuous bag-of-words and skip-gram. word2vec was developed by Mikolov, Sutskever, Chen, Corrado and Dean at Google and published in 2013 [36]. The two model architectures were made available as an open-source toolkit written in C [37].

word2vec takes a text corpus as input and produces word vectors as output. word2vec is using a neural network and deep learning, even though the network is relatively shallow in comparison to other approaches [65]. The plain softmax word2vec essentially counts how many times words occur together [65]. Mathematically, Levy and Goldberg derived the word2vec models as implicit factorizations of the shifted positive pointwise mutual information matrix (SPPMI) [16]. Levy et al. describe word2vec as an efficient implementation of decade-old ideas and explain that much of the improvement in performance stems from pre-processing and hyperparameter settings [32].

3.3.1 Continuous bag-of-words

The continuous bag-of-words (CBOW) architecture predicts the current word given its context, e.g. by taking the input words w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2} to output w_i. Both the words left and right from the current word are taken into account when making the predictions. With CBOW, the order of words in the history does not influence the projection. According to Mikolov, this approach is faster to train and more appropriate for large corpora [36].

3.3.2 Continuous skip-gram

The continuous skip-gram architecture predicts the context given the current word. The input w_i is used to output w_{i-2}, w_{i-1}, w_{i+1}, w_{i+2}. It is called skip-gram, because it can also skip some of the words in the context, e.g. by using w_{i-4}, w_{i-1}, w_{i+1}, w_{i+4}. Practically, the current word is used to predict the surrounding words. According to Mikolov, this approach results in better word vectors for frequent words as well as better word vectors for infrequent words, even though it is slower to train than CBOW [36].

3.3.3 Parameters

If trained long enough, CBOW and skip-gram perform similarly. In addition to that, the vectors can be trained with different training algorithms like hierarchical softmax and negative sampling. Hierarchical softmax is better for infrequent words, negative sampling is better for frequent words and low dimensional vectors [36].

For the vector dimensionality, Mikolov et al. say that more is better [36]. For skip-gram, they recommend a context window size of around 10, for CBOW they recommend a context window of around 5 words. The initial training time of the Collobert NNLM with N=5 dimensional vector and 660 million training words was 2 months.

3.4 GloVe

Global Vectors (GloVe) is an unsupervised learning algorithm trained on aggregated global word-word co-occurrence statistics from a corpus. GloVe explicitly identified the objective that word2vec optimizes and connected it to the well-established field of matrix factorization [65]. GloVe identified a matrix that, when factorized using the same stochastic gradient descent (SGD) algorithm as word2vec, results in a word and a word context matrix [46]. In contrast to word2vec, GloVe explicitly names the objective matrix, identifies the factorization, and provides some intuitive justification as to why this yields working similarities [65].

3.5 Dependency-based word embeddings

Dependency-based word embeddings describe an alternative training algorithm where the context is based on the syntactic relation the word participates in [31]. This helps to capture relations between words that are far apart and would be "out-of-reach" for a bag-of-words approach with a limited context window.

As explained by Goldberg and Levy, the word2vec approach finds words that associate with a word (domain similarity) while the dependency-based approach finds words that behave like a word (functional similarity) [31]. Generally, the bag-of-words approach reflects the domain aspect while the dependency-based approach captures the semantic type of a target word (see Figure 5).

For example, "Hogwarts", the school in the Harry Potter universe, is associated with other themes and characters from the Harry Potter universe for the bag-of-words approach with a 5-word context window. The dependency-based approach yields a list of other famous schools like Sunnydale from the Buffy the Vampire Slayer universe. The word2vec bag-of-words approach associates Florida with cities in the state of Florida like Jacksonville and Lauderdale, while the dependency-based word embeddings associates Florida with other U.S. states like Texas and California.

3.6 Random Indexing

Random Indexing is an incremental word space model that avoids constructing a huge co-occurrence matrix and that works well with sparse input [48]. In Random Indexing, each term is represented by a n-dimensional random context vector. For most of the n dimensions, the vector is set to 0. For a small number k, the vector has randomly distributed -1 or +1 values. For instance, n=1800 and k = 8.7 [24]. For each observed occurrence of a word w_i, the vectors of the words in the context window $v_{i-3}, \ldots, v_{i+3}$ are added to the word's context vector v_i. After a number of occurrences, the context vector holds information about a word's distribution.

The context vectors are in a fixed-dimensional space of comparatively low dimensionality [48]. More generally, random mapping was shown as a promising and computationally feasible alternative for dimensionality reduction that is computationally less costly than methods like principal component analysis [26].

Target Word	BoW5	BoW2	Deps
batman	nightwing	superman	superman
	aquaman	superboy	superboy
	catwoman	aquaman	supergirl
	superman	catwoman	catwoman
	manhunter	batgirl	aquaman
hogwarts	dumbledore	evernight	sunnydale
	hallows	sunnydale	collinwood
	half-blood	garderobe	calarts
	malfoy	blandings	greendale
	snape	collinwood	millfield
turing	nondeterministic	non-deterministic	pauling
	non-deterministic	finite-state	hotelling
	computability	nondeterministic	heting
	deterministic	buchi	lessing
	finite-state	primality	hamming
florida	gainesville	fla	texas
	fla	alabama	louisiana
	jacksonville	gainesville	georgia
	tampa	tallahassee	california
	lauderdale	texas	carolina
object-oriented	aspect-oriented	aspect-oriented	event-driven
	smalltalk	event-driven	domain-specific
	event-driven	objective-c	rule-based
	prolog	dataflow	data-driven
	domain-specific	4gl	human-centered
dancing	singing	singing	singing
	dance	dance	rapping
	dances	dances	breakdancing
	dancers	breakdancing	miming
	tap-dancing	clowning	busking

Figure 5: Different word representations and what semantic similarity they encode for different approaches. From left to right: word2vec bag-of-words with context windows of 5 (BoW5) and 2 (BoW2) and dependency-based word embeddings (DEPS) [31].

3.7 Which representation to choose

For the topic comparison purposes of this thesis, the domain similarity captured by word2vec was preferred over the functional similarity captured by the dependency-based approach. Therefore, a choice had to be made between the models providing domain similarity. There is a trade-off between using more memory (GloVe) and taking longer to train (word2vec) [65]. According to Mikolov, the choice of training corpus is usually more important than the choice of technique itself [36]. The open-source word2vec C tool released by Google and the Python bindings available in gensim were used [37] as this opened the possibility to use the freely available word vectors that were trained on a Google data set with 100 billion words.

4 Implementation

4.1 Modules

A variety of different modules was developed during my internship. In this thesis, the Topic and Topic Comparison modules are described. Each module will be described and discussed in regard to what it does, how it relates to theory, how it is implemented, the motivation for the approach and self-criticism.

4.1.1 Topics module

Description The Topic Module provides a bird's-eye view of a text. It takes a text and maps the words so that semantically and stylistically similar words are close to each other (see Figure 6). This enables users to explore a text source like a geographical map. As similar words are close to each other, the user can visually identify clusters of topics that are present in the text. Conceptually, it can be understood as a "Fourier transformation for text".

Theory As discussed in Section 3.1, word vectors capture many linguistic properties such as gender, tense, plurality and even semantic concepts like *is capital city of*, which we exploit using a combination of dimensionality reduction and data visualization.

Implementation The topic module implements the following steps:

0. Pre-processing In the pre-processing step, all sentences are tokenized to extract single words. The tokenization is done using the Penn Treebank Tokenizer implemented in the Natural Language Processing Toolkit (NLTK) for Python [5]. Alternatively, this could also be achieved with a regular expression.

Using a hash map, all words are counted. Only unique words, i.e. the keys of the hash map, are taken into account for the dimensionality reduction. Not all unique words are taken into account. The 3000 most frequent English words according to a frequency list collected from Wikipedia are ignored to reduce the amount of data [59].

1. Word representations For all unique non-frequent words, the word representation vectors are collected from the word2vec model via the gensim Python library [66]. Each word is represented by an N-dimensional vector (N=300).

2. Dimensionality Reduction The results of the word2vec vectors are projected down to 2D using the t-SNE Python implementation in scikit-learn (See Figure 7) [45].

(a) Linguistics (b) Computer Science

(c) Mathematics (d) Business

Figure 6: Different clusters after word2vec and t-SNE.

3. Visualization After the dimensionality reduction, the vectors are written to a JSON file. The vectors are visualized using the D3.js JavaScript data visualization library [6]. Using D3.js, an interactive map was developed. With this map, the user can move around and zoom in and out.

Motivation There are a variety of different ways to approach the problem of visualizing the topics in a text. The simplest way would be looking at unique words and their occurrences and visualizing them in a list. The topics could also be visualized using word clouds, where the font size of a word is determined by the frequency of the word. Word clouds have a variety of shortcomings: They can only visualize a small subsets, the focus on the most common words is not helpful for the task at hand and they do not take synonyms and semantically similar words into account.

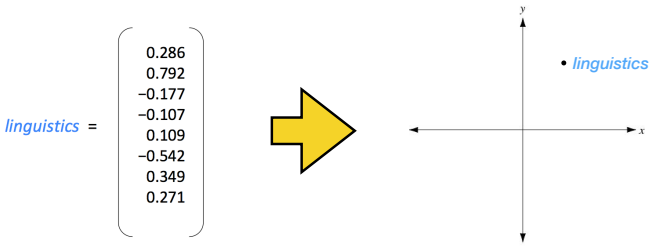


Figure 7: In the dimensionality reduction step, the 300 dimensional word vectors are projected down to a two–dimensional space, so that they can be easily visualized in a 2D coordinate sytem.

Therefore, the bird's-eye view approach was developed and favoured. Section 3.7 details why the word2vec implementation [36, 37] was used instead of Random Indexing [48] or the dependency-based word embeddings from Levy and Goldberg [31].

To develop an intuition for the approach, imagine a set of words such as *disorders*, *neurological*, *obesity*, *nutrition*, *diet* and *pathological craving*. For a human judge, it is easy to understand what topic these words might be related to. Likewise, a set of words like *purchase*, *profits*, *employees*, *marketing*, *customer*, *consumers*, and *loyalty*, can be easily understood as a set of words that describes phenomena from the business world. Another region might consist of words like *laboratory*, *studies*, *experimental*, *experiments*, *researchers*, and *scientists*.

As the vectors encode semantic and syntactic similarity, similar words are encoded and visualized close to each other. If there are many semantically similar words in a text, this leads to the formation of regions.

Self-criticism It would be good to have the ability to remove certain words or fold them up into labelled clusters. This would allow the users to reorganize and simplify the visualization output. The results could be automatically colour-coded, e.g. based on their WordNet categories [40].

Another useful addition would be a comment feature, that would enable editors to discuss the topics within the application. The topic module is also missing a direct link to the occurrences of the words in the text.

4.1.2 Topic Comparison module

Description The Topic Comparison module can be used to compare a summary and its source material. It extends upon the Topic module and uses the same tool chain described in the previous Section 4.1.1.

To compare the topics, three different sets of words are computed: a source text topic set, a summary topic set, as well as the intersection set of both topic sets (see Figure 8). These three sets are then visualized similarly to the Topic module. A colour is assigned to each set of words. This enables the user to visually compare the different text sources and to see which topics are covered where. The user can explore the word map and zoom in and out. He or she can also toggle the visibility, i.e. show and hide, certain word sets.

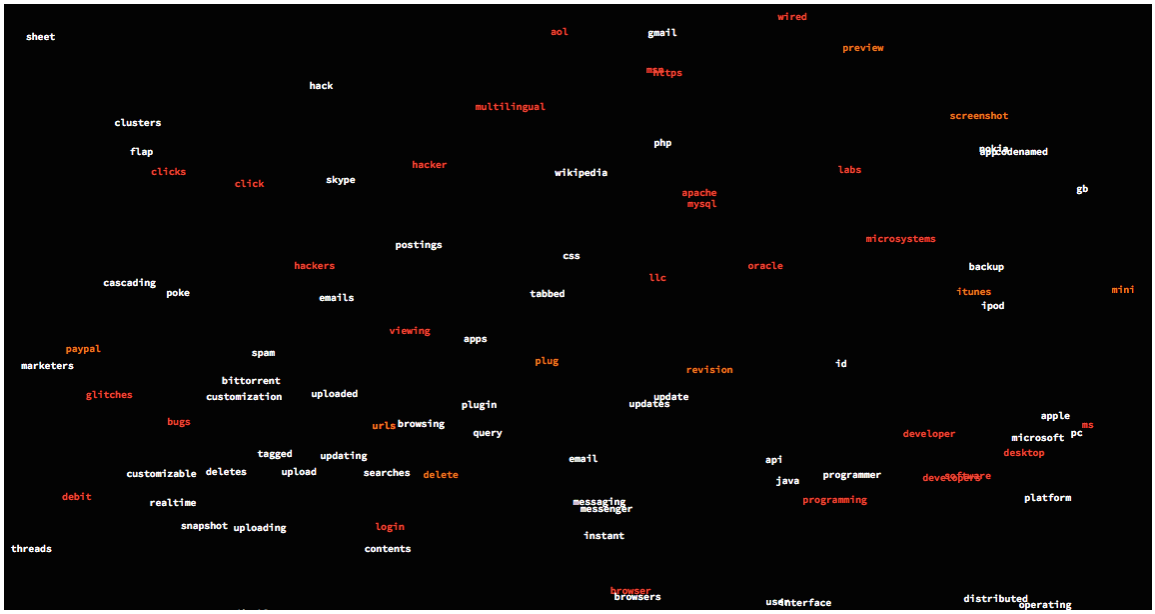


Figure 8: Topic Comparison module with the Topics A (orange), Topics B (red) and the intersection of Topics (white).

Theory One of the goals of this thesis was to find a way to assess a summary and its source material in regards to the number of examples and the number of stories. As this is a highly subjective task that is hard to automate, the tool took a user experience and human-computer interaction-inspired approach to provide a novel way of comparing two text sources.

When summarizing a large text, only a subset of the available stories and examples can be taken into account. The decision which topics to cover is largely editorial. The Topic Comparison module assists this editorial process. It enables a user to visually identify agreement and disagreement between two text sources.

Implementation The Topic Comparison module shares a lot of code with the Topic Module, which is described in Section 4.1.1. As input, it uses the three sets of

words and renders them with different colours and visualizes them in a D3.js-based interactive map, where the user can zoom in and out, move around, and toggle the different groups.

Both the frontend and the backend of the implementation were made available on GitHub under GNU General Public License 3 [20]. The repository includes the necessary Python code to collect the word2vec representations using Gensim, to project them down to 2D using t-SNE and to output them as JSON. The repository also includes the frontend code to explore the JSON file as a geographical map.

The Github repository also includes an online demo of the tool [20]. The tool can be used to explore the precomputed topic sets of the Game of Thrones Wikipedia article revisions from 2013 and 2015. The repository also includes the precomputed topic sets for the Wikipedia article revisions for the articles on World War 2, Facebook, and the United States of America.

Motivation The visualization automatically highlights regions of words. By looking at the intersection set, the user can immediately see what is covered in the summary. Comparing two sets of words numerically would be a $\mathcal{O}(n^2)$ problem with a very large n.

As the comparison of the texts and the source material can be quite complex and hard to oversee, different visualization techniques were sketched and evaluated by ad-hoc user tests. In the initial design, the words were represented simply by dots. Users were able to hover over the dots to see the word and other additional information. This was replaced by a rendering of the word itself. This, however, increased the complexity as there were strong clusters of words where the text was unreadable. Therefore, the tool provides an option to change both the zoom factor and the scale factor, i.e. the spread of the words.

Self-criticism The implementation uses t-SNE, which minimizes the Kullback-Leibler divergence between the joint probabilities of the high-dimensional data and the low-dimensional representation. Therefore, the two text sources need to be trained simultaneously so that the vector spaces are aligned after the dimensionality reduction step. It would be preferable to store each source text individually and be able to compare them ad-hoc, especially since t-SNE is computationally expensive. It would also be beneficial to include a numerical measure of how similar or dissimilar a text source and a summary are.

5 Experiments and results

5.1 Methodology

For this thesis, the research approaches were evaluated using prototypes and ad-hoc user tests. As the primary objective was to explore the state-of-the-art in research and aid the quality assurance and quality control efforts of a company, a formal evaluation of the findings was out of scope for the thesis work.

The requirements of this thesis were determined by operationalizing the use case of a company. For this, user observations and user interviews were conducted. Based on this, a literature review was compiled.

To empirically show the effectiveness of the approach and to make a final assessment of the usefulness of the developed tools, further user tests, and user studies are required. This section outlines possible experiments to empirically evaluate the developed tools.

For a satisfactory evaluation of the developed tools, each module would require a thorough evaluation. The use case could benefit from a more normative approach that combines task analysis, heuristic evaluations, and usability testing.

5.1.1 Task analysis

To evaluate the modules, it would be beneficial to apply a technique like hierarchical task analysis (HTA). The goal of task analysis is to decompose complex human activities into tasks. Task analysis methods usually include hierarchies, sequences, and choice. In a hierarchy, task B is a subtask of task A. In a sequence, task A follows task B. In regards to choice, the user has to choose between task A and task B [2].

For the Topic module and the Topic Comparison, an HTA could be used to evaluate how easily a user can use the tool to explore a text source and decide which topics to include and which to omit.

This could then be evaluated in a variety of ways. Experts could evaluate the revisions of Wikipedia articles. The users could be interviewed afterward on their experience using the tool. The process itself could also be evaluated via usability testing.

5.1.2 Heuristic evaluation

In a heuristic evaluation, an expert, i.e. somebody trained in HCI and interaction design, examines a proposed design to see how it measures up against a list of principles, guidelines or *heuristics* [4]. This can cover everything from aesthetics to human error and is often backed up by psychological theories and empirical data [47].

For this thesis and its goal, Nielsen's *10 Usability Heuristics for User Interface Design* would for instance be unsuited, as it focuses on aspects like error prevention, the visibility of system status, and consistency and standards [42]. However, the flexibility and efficiency of use would be an important heuristic to evaluate.

5.1.3 Usability testing

An interactive prototype can be evaluated against in-house guidelines as well as formal usability standards such as ISO 9241 [4]. The guidelines should be evaluated through user tests. ISO 9241-11 defines usability as "the extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use" [15]. However, the variables effectiveness, efficiency, and satisfaction are hard to apply to high-level concepts such as how well an editor can compare Wikipedia article revisions.

5.1.4 Topic and Topic Comparison module evaluation

As the Topic and Topic Comparison module can be regarded as the main contribution of this thesis, a more thorough evaluation of these modules will be outlined here. To evaluate the Topic and the Topic Comparison modules, a data set of text summaries or different revisions of text sources Wikipedia articles could be used.

A group of experts could rank the most salient topics in a text. This could serve as a ground truth, which could then be modified by removing or adding topics by hand. The goal of the user test could be to see if users benefit from the Topic and Topic Comparison modules when fulfilling a topic detection task. After the user tests, the users could do a survey to rank how satisfied they are with the experience. Open-ended questions could assess what problems they encountered. These surveys could be supplemented by qualitative interviews.

Efficiency could be measured by looking at the time it takes the users to decide which topics to have been added or removed from a Wikipedia article revision. This could be compared to a second group of users which do not use the Topic and Topic Comparison modules for the task.

Effectiveness would be the hardest to quantify. It could be evaluated based on the topics that are selected and how they are ranked. This could be compared to a ground truth. For this, it might be beneficial to do an expert evaluation to find a ground truth. However, even this can only be based on intersubjective agreement, especially considering that text sources can be hundreds of pages long and that they feature a variety of different topics. The size of the sample set should reflect that this is a very individual process, where the time measurement may vary significantly between different users.

5.2 Used data

5.2.1 Topic comparison

To evaluate and test the Topic Comparison module for this thesis, Wikipedia articles were used as a proxy. For this, the flow described in Section 4.1.1 was applied to different revisions of Wikipedia articles. A convenience sample of the most popular articles in 2013 from the English Wikipedia was used. The list is ranked by view count [18]. For each article, the last revision from the 31st of December 2013 and the most recent revision on the 26th of May 2015 were collected. The assumption was that popular articles will attract sufficient changes to be interesting to compare.

A convenience sample was used as articles such as "Deaths in 2013" or "List of Bollywood film 2013" are not very useful for the task of comparing the topics in a text as they did not change over time.

The list of the most popular Wikipedia articles includes Facebook, Game of Thrones, the United States, and World War 2. Especially the article on Game of Thrones was deemed useful for the task of comparing the topics in a text, as the storyline of the TV show developed between the two different snapshot dates as new characters were introduced. Other characters became less relevant and were removed from the article. The article on World War 2 was especially interesting as one of the motivations for the Topic Comparison module is to find subtle changes in data.

For this thesis and its experiments, word vectors trained on the small 100 MB text8.gz text corpus from the web provided by Google were used [36]. It would have been beneficial to not only use the word vectors trained on the small demo dataset, especially since Mikolov et al. provide 1.4 million pre-trained entity vectors that were trained on 100 billion words from various news articles [37]. Custom word vectors trained on a large domain-specific dataset, e.g. a large corpus of documents, would have been even better.

5.2.2 LDA topic detection

Topic modelling is a very time-consuming and computationally heavy process that requires large text corpora and a lot of computing time. Řehůřek shows that performing LDA on the English Wikipedia requires a lot of free disk space (35GB) and a long pre-processing time (9 hours for the English Wikipedia) [64]. Creating the LDA model of Wikipedia itself takes 6 hours and 20 minutes [64]. In "Macroanalysis: Digital Methods and Literary History", Jockers used LDA to extract 500 themes from a corpus of 19th-Century Fiction [22]. Jockers assigned labels to topic clusters in a subjective process. For each topic, a word cloud is available with words associated to the theme. Jockers' data is available as JSON and was used as a starting point for a simple count-based topic detection system [23].

5.2.3 Text quality gold standard

Section 5.4 describes how to compute a text quality gold standard using machine learning. For these experiments, 14 transcripts of the podcast *This American Life* produced by the National Public Radio (NPR) from the USA were collected. This includes the episodes 537 to 551. *This American Life* is a weekly public radio show with about 2.2 million listeners. As a podcast, it has around one million downloads per week, which, on their own account, makes them the most popular podcast in the USA.

5.3 Topic visualization and comparison

One key aspect of this thesis was to explore how the topic comparison of large text sources can be supported using natural language processing and machine learning. In this context, a topic can be defined as any word that is present in a text.

Topics in a text source are not always represented by nouns but can also be represented by certain verbs like *analyzing* or certain adjectives like *beautiful*. Therefore, a universal approach is needed to identify and compare different source texts with an abundance of topics. Considering four randomly chosen books from a corpus of books, the number of words in each book is relatively high: 18966, 39789, 74810, 85740. Even though there are much less unique words in the books, processing them in a meaningful way is hard. Especially if every word in one input source needs to be compared to every word in another input source.

To mitigate this, this thesis explored a novel visualization approach. Instead of looking at statistical measures like word occurrences, the visualization approach used word vector representations and dimensionality reduction to project them into a 2D coordinate system.

Using this approach, visual clusters of words emerge, which enable editors to easily identify topical clusters of words. As similar words are close to each other after the t-SNE 2D projection, the regions in the 2D space represent topics. Rather than providing a list of word occurrences or bags-of-words like in LDA, this approach is more like a geographical map.

This yields a novel and universal way of processing large collections of text. Intuitively, it serves as a "Fourier transformation for text" and results in a bird's-eye view on a text. The user can zoom in and out and move around. This enables the user to have both, a global overview (zoomed out) and a local detail view (zoomed in on a specific region).

When different source texts are plotted in the same map, they can be compared visually. There could be a cluster of words related to banking for one text source while another text source, e.g. the summary, would not feature these words. This might indicate that the text does include a story about banking, which is not present in the summary.

5.3.1 Topic comparison of Wikipedia revisions

In the following sections, the Topic Comparison module described in Section 4.1.1 will be applied to different revisions of Wikipedia articles described in Section 5.2.1 to demonstrate how the module exposes regional clusters, global clusters, and how it facilitates topic comparison.

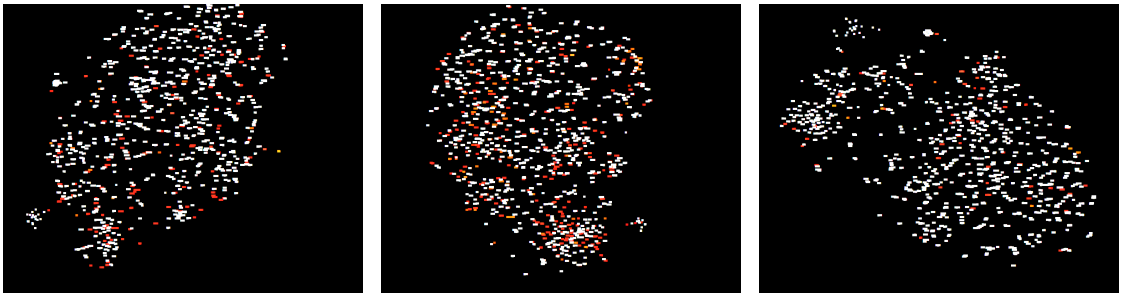
5.3.2 Regional cluster

Figure 9 shows a regional cluster of words in the Wikipedia article on Game of Thrones related to television and acting.

Figure 9: Game of Thrones: Semantically and stylistically similar words end up being close to each other.

5.3.3 Global clusters

Figure 10 shows the articles of three Wikipedia articles and their revisions from 2013 and 2015 including the article on the United States of America, Game of Thrones and World War 2.

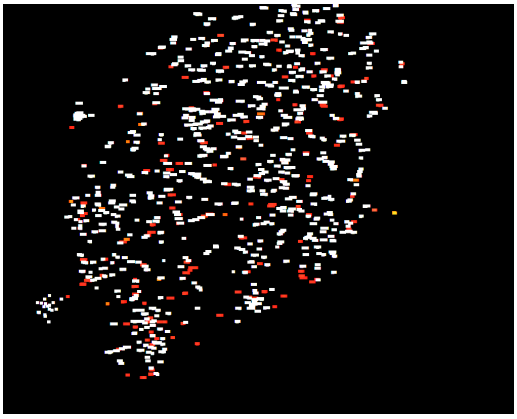


(a) United States (b) Game of Thrones (c) World War 2

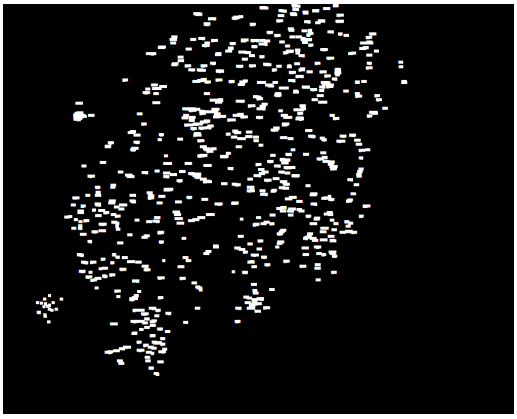
Figure 10: Topic Module bird's-eye view of three Wikipedia articles and their revisions from 2013 and 2015.

5.3.4 Topic comparison I

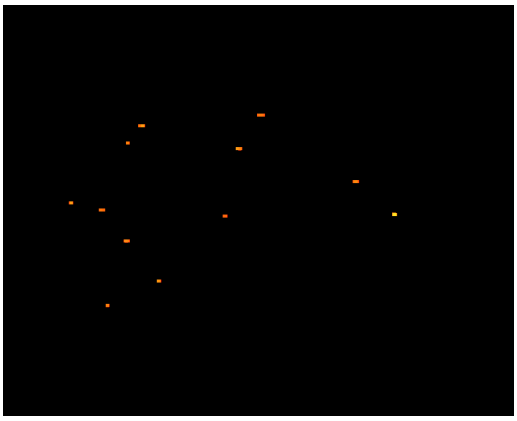
Figure 11 shows how an editor would view all sets, only the intersection set, the set of words only present in the 2013 revision and the set of words only present in the 2015 revision of the Wikipedia article revision about the United States.



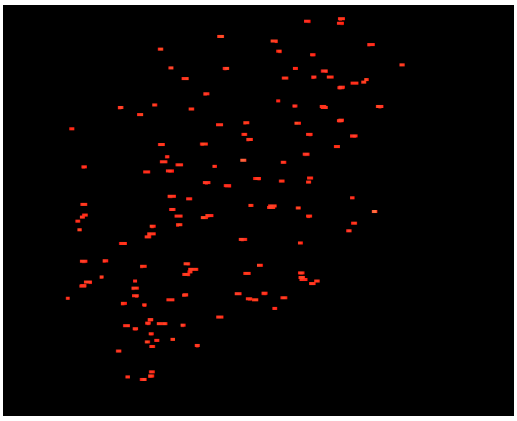
(a) All words



(b) Intersection set



(c) 2013 revision

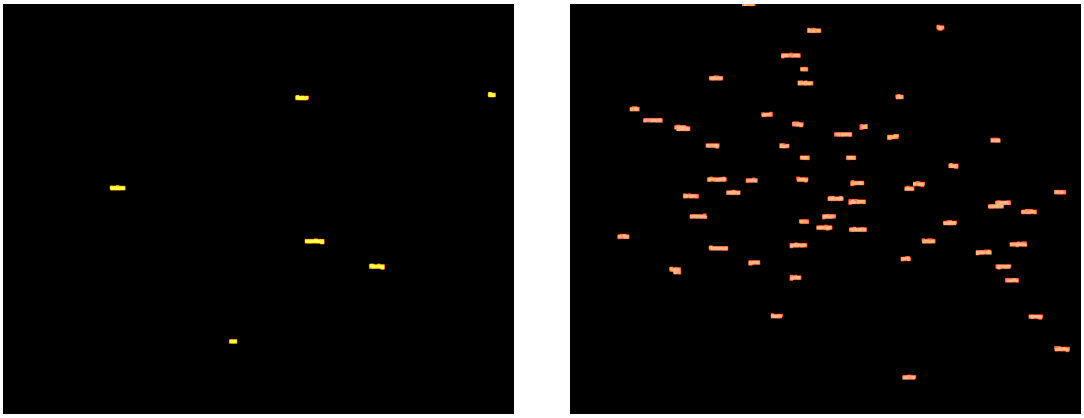


(d) 2015 revision

Figure 11: Topic Comparison module visualizing the Wikipedia article about the United States.

5.3.5 Topic comparison II

Figure 12 compares the Game of Thrones Wikipedia article revisions in regards to character names. Figure 12a) shows that a few characters were removed from the article and are only present in the 2013 revision. Figure 12b) shows that a variety of character names were added to the article in 2015.

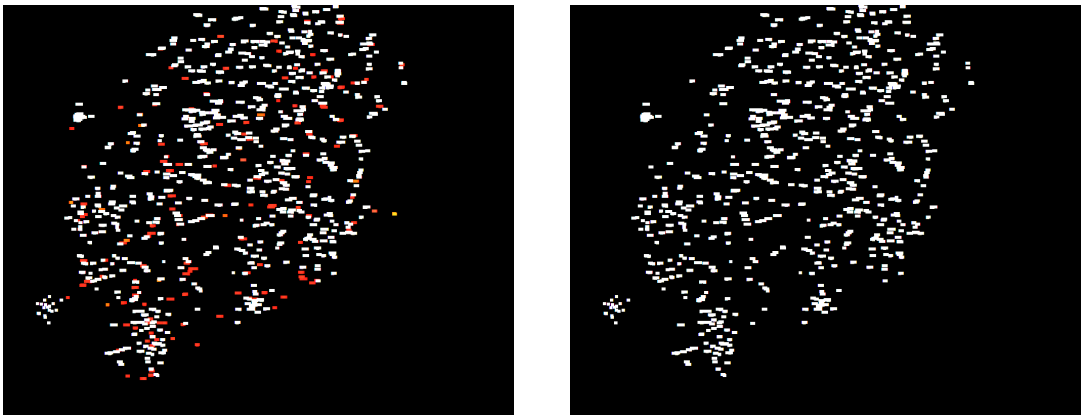


(a) 2013 revision (b) 2015 revision

Figure 12: Comparison of character names in Game of Thrones article.

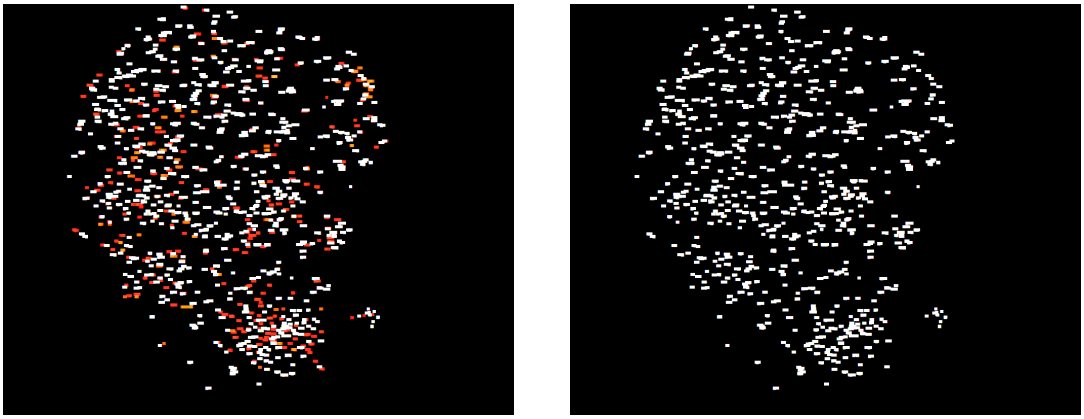
5.3.6 Intersection sets

The Figures 13-15 compare the intersection sets of words present in both the 2013 and the 2015 revisions of the Wikipedia articles on the United States (Figure 13), Game of Thrones (Figure 14) and World War 2 (Figure 15).



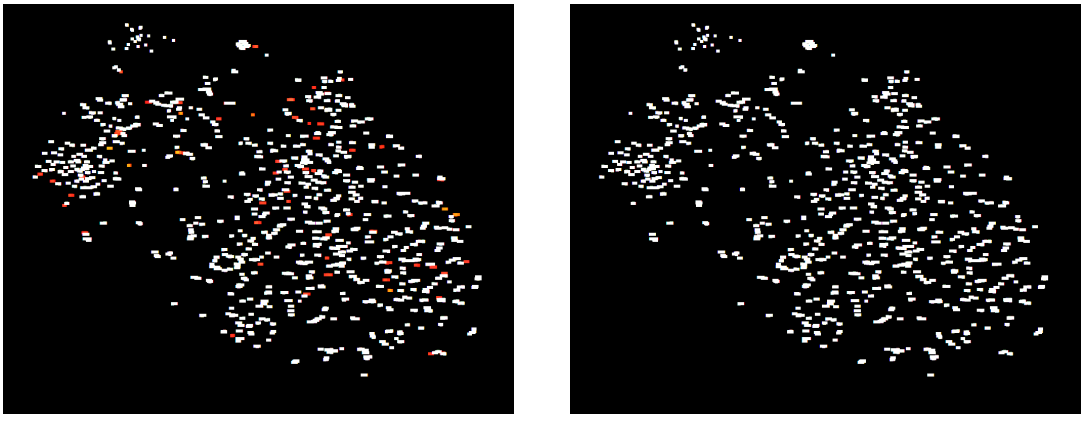
(a) All words (b) Intersection set

Figure 13: United States of America



(a) All words (b) Intersection set

Figure 14: Game of Thrones



(a) 2013 revision (b) 2015 revision

Figure 15: World War 2

5.4 Text quality gold standard

Text quality can be described as a multi-faceted concept, which is hard to formalize a priori, but which can emerge a posteriori from machine learning, which fundamentally is pattern recognition. The desired text quality is a pattern which can be derived from a set of examples.

One of the main incentives for the development of the platform is quality assurance and quality control. Aspects like the level of readability or emotionality that are desired to be in the text are hard to define by hand. They can, however, emerge from analysing existing documents, which have been approved by human judgment. Therefore, the task can be operationalized as a supervised machine learning task.

To solve this supervised machine learning problem, a variety of different approaches can be used and a variety of metrics can be taken into account. The task can be treated as a) a regression problem, where the goal is to predict a numerical quality rating, e.g. a label that describes the quality of a text on a discrete scale, or b) a binary classification problem, where texts are classified as high quality and low quality texts, or c) to derive a linear or logistic regression model that predicts the best values for certain sections based on seen data. In the following, approach c) will be implemented.

5.4.1 Input features

To train a machine learning classifier, input features need to be defined. For this application, the Flesch-Kincaid Score, the Fog Score, as well as the Smog Score per text chunk were used.

As described in Section 5.2.3, the transcripts of the podcast *This American Life* were used as a proxy for this experiment. For each section of 30 sentences, the Flesch-Kincaid Score, the Fog Score as well as the Smog Score were computed. The transcripts include a variety of short lines that only consist of character names. To prevent this from influencing the readability scores, all sentences that consist of less than two words were dropped and did not influence the rating of the readability scores.

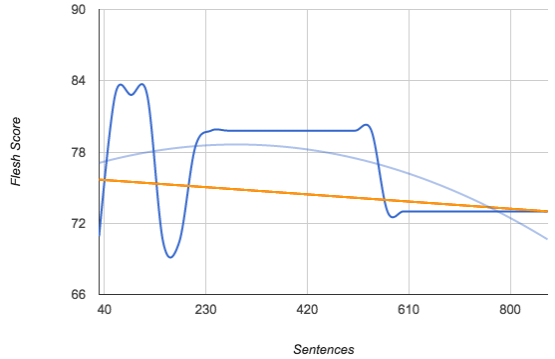
5.4.2 Regression model

Using the input features, a linear and a logistic regression model were derived to predict the best readability values. The values can serve as a model summary, representing an implicit gold standard.

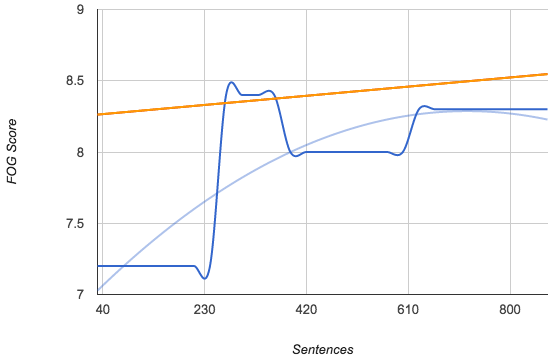
The linear model represents the global readability, which is a constant readability across the entire text. The goal is to have a homogeneous text with little variation over time.

The logistic model represents the local readability and assigns each section a desired readability score. The goal is to find a fingerprint, which closely resembles a model summary.

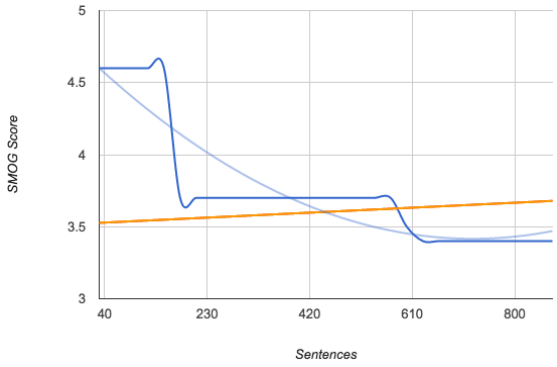
For each of the three different readability scores, a machine learning model was fitted with a Logistic Regression (blue) as well as a Linear Regression (orange for Ordinary Least Squares, red for Lasso). Figure 16a) shows the Flesch-Kincaid gold standard, Figure 16b) shows the FOG gold standard and Figure 16c) shows the SMOG gold standard. For each graph, a trend line was computed (light blue).



(a) Flesch



(b) FOG



(c) SMOG

Figure 16: Prediction based on This American Life with Logistic Regression (dark blue), Trend line (light blue), and Linear Regression (red, orange).

Figure 17 plots the three curves against each other. This helps to gain an intuition about the reading scores and how they interact and what they are sensitive to. The gold standard can then be compared to any text to assess how similar a text source is to the desired readability. Figure 18 shows a comparison of the Wikipedia article on Facebook and the SMOG readability gold standard.

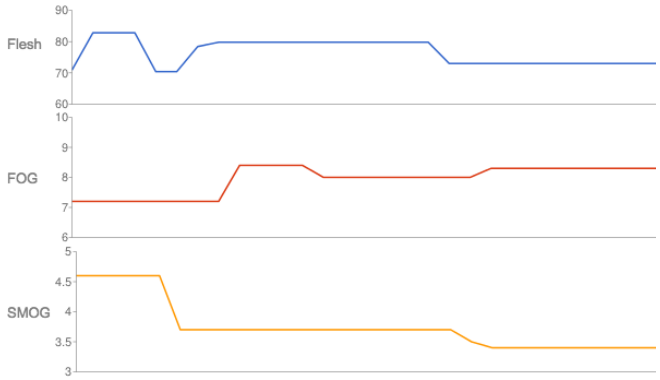


Figure 17: Reading Score Comparison: Flesch-Kincaid (blue), FOG (red), and SMOG (orange).

A similar model could be trained for the emotionality. Here, the eight dimensions of emotionality computed by the Gavagai Sentiment API could be used as an input.

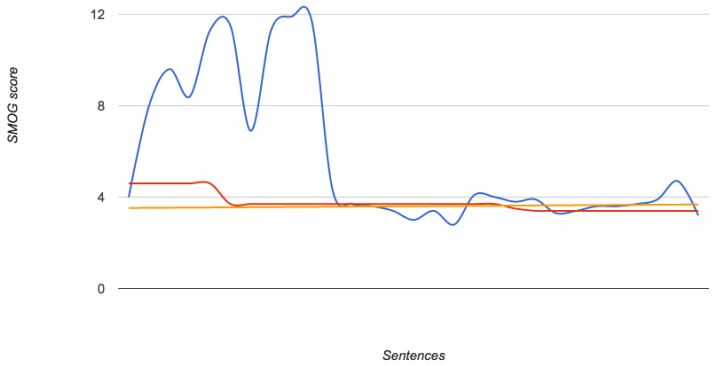


Figure 18: Wikipedia article on Facebook compared to SMOG gold standard: Facebook Wikipedia SMOG score (blue), logistic gold standard (red), and linear gold standard (orange).

5.4.3 Conclusions

This experiment showed how machine learning can be used to derive a model curve for a certain property like readability. Using this model, a gold standard of readability change over time for *This American Life* was computed. This can be used to evaluate new transcripts in regards to how similar they are to a *This American Life* gold standard. A draft of a new episode can be compared to this gold standard. The editors can visually identify text sections that are too different from the gold standard and improve them.

6 Conclusions

For this thesis, the use case of a company was operationalized and a variety of different modules was proposed, implemented and tested. Natural language processing and machine learning techniques were applied to aid the quality control and quality assurance efforts of a company.

The main contribution of this thesis is a novel way of doing text comparison using word vector representations and dimensionality reduction. Word2vec word vector representations and t-SNE dimensionality reduction are used to provide a bird's-eye view of different text sources, including text summaries and their source material. This enables users to explore a text source like a geographical map. Semantically similar words are close to each other in 2D, which yields a "Fourier transformation for text".

A simple regression model is introduced and applied to derive a gold standard from a collection of texts.

The main goal of the thesis was to support the quality control and quality assurance efforts of a company. This goal was operationalized and subdivided into several modules.

For each module, the state of the art in natural language processing and machine learning research was investigated and applied. The implementation section of this thesis discusses what each module does, how it relates to theory, how the module is implemented, the motivation for the chosen approach and self-criticism.

6.1 Advantages of this approach

The thesis investigated the applicability of recent research results. As many researchers publish their source code under open source licenses, it was possible to integrate the findings from the literature review into a useable tool.

The most innovative aspects of this thesis work are the Topic module and the Topic Comparison module, which address a complex problem – comparing two text sources with each other – using word representations, dimensionality reduction and data visualization. The Topic Comparison tool developed for this was made available under GNU General Public License 3 [20]. The Story module is innovative as it uses a simple word occurrences approach to detect and visualize how the stories in a text change over time.

Machine learning techniques are used to automatically derive a gold standard for indicators like readability or emotionality. These indicators can help guide human judges to improve the text quality and enable users to come as close as possible to a certain text quality and text style.

6.2 Disadvantages of this approach

The major flaw of the thesis is that the introduced text visualization and text comparison approach is not validated empirically. While this was inevitable giving the scope and the complexity of the thesis and the problems at hand, it would have

been much more satisfying to describe and develop a single big contribution that can quantify text style similarity. Many of the unpublished modules are too simplistic and rely too much on word occurrences and only provide proxies for complex problems.

In hindsight, it might have been better to focus on a specific practical or theoretical problem and address it using a novel technique with a thorough empirical evaluation.

The thesis also did not make enough progress on stylistic text analysis. Initially, the hope was to be able to work on the problem of comparing one text stylistically to another. The thesis entails work to facilitate this and achieves it for individual parts. However it would have been more satisfying to develop a single machine learning system, e.g. a deep neural network, that would have been able to use these different features and take two summaries as an input to predict how similar they are in regards to their style.

7 Future work

During the implementation and documentation of this thesis work, various publications introduced novel approaches. Due to time limitations, not all could be explored and implemented as part of this thesis. The following sections discuss possible future work with a special focus on deep learning.

7.1 Sentence and document vectors

For the purpose of text summarization, topic comparison, information extraction and topic modelling, it would have been beneficial to obtain sentence level representations. There has been a lot of work on sentence and document level representations similar to the skip-gram and CBOW models described for words. The progress on paraphrase detection could be used to improve the topic comparison [52]. Socher showed that unfolding recursive autoencoders (URAE) can not only capture and memorize single words but also longer, unseen phrases [52]. Socher also showed that URAE can learn compositional features beyond the initial word vectors and identify the most complex paraphrase relationships to improve accuracy [52].

According to Le and Mikolov, sentence vectors provide state of the art results on sentiment analysis tasks, even beating approaches such as recursive neural networks [29]. Sentence and document vectors like doc2vec could be used to compare sentences and paragraphs to each other.

7.2 Automatic compliance assessment

Given sufficient input data, the system could be extended to provide a single model to assess the compliance to a certain text style or a stylistic gold standard. The goal would be to assess how similar a draft is to this archetypical summary. For this, certain features could be used to train a neural network, which could then assess and quantify how similar a text is to a desired text style.

Producing summaries that both reproduce the content of a text accurately and that follow the same stylistic format might be hard to fully automate. The content aspect could be addressed with a semi-automated approach. Graph-based algorithms can accurately summarize texts [11, 12]. They can provide a good starting point on what topics to focus on.

7.3 Deep learning

Deep Learning could also be used for the task of semantic and stylistic text analysis and text summary evaluation. Neural networks are universal approximators. Any neural network with nonlinearities can represent any function, in the worst case by simply acting as a lookup table. Deep learning with many hidden layers brings performance improvements and improves generalization. To use deep learning, an objective function needs to be defined that can be used during training to reduce

the training error. Anything that can be formalized can be used as an objective function to train the neural network.

Deep learning approaches like the Neural Turing machine or other memory networks can be used for tasks that require reasoning and symbol manipulation [30]. LeCun et al. provide the example of a network that is shown a 15-sentence version of the *The Lord of the Rings* and that correctly answers questions such as "Where is Frodo now?" [30].

Recurrent Neural Networks (RNN) are already able to accomplish machine translation tasks. For this, an English sentence is read one word at a time to train an English *encoder network* so that a hidden state vector is a good representation of the thought expressed in the English sentence. This *thought vector* is then used to initialize a French *decoder network*, which provides a French translation [30].

With these limitations in mind, the problem could be classified as a *strong AI problem*, i.e. a problem that requires artificial general intelligence (AGI) to successfully perform an intellectual task that a human being with sufficient training can perform.

8 References

[1] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau. Sentiment Analysis of Twitter Data. In *Proceedings of the Workshop on Languages in Social Media*, LSM '11, pages 30–38, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[2] John Annett. Hierarchical task analysis. *Handbook of cognitive task design*, pages 17–35, 2003.

[3] Yoshua Bengio, Réjean Ducharme, Pascal Vincent, and Christian Janvin. A Neural Probabilistic Language Model. 3:1137–1155, March 2003.

[4] David Benyon. *Designing Interactive Systems: A comprehensive guide to HCI and interaction design*. Pearson Education Limited, 2010.

[5] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media, Inc., 1st edition, 2009.

[6] Mike Bostock. D3.js - data-driven documents. `http://d3js.org/`, 2012.

[7] Elia Bruni, Nam Khanh Tran, and Marco Baroni. Multimodal Distributional Semantics. *J. Artif. Int. Res.*, 49(1):1–47, January 2014.

[8] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural Language Processing (Almost) from Scratch. *J. Mach. Learn. Res.*, 12:2493–2537, November 2011.

[9] John Conroy and Dianne P. O'leary. Text Summarization via Hidden Markov Models and Pivoted QR Matrix Decomposition. Technical report, In SIGIR, 2001.

[10] Hermann Ebbinghaus. *Memory: A contribution to experimental psychology*. Teachers college, Columbia university, 1913.

[11] Günes Erkan and Dragomir R. Radev. LexRank: Graph-based Lexical Centrality As Salience in Text Summarization. *J. Artif. Int. Res.*, 22(1):457–479, December 2004.

[12] Debora Field, Stephen Pulman, Nicolas van Labeke, Denise Whitelock, and John Richardson. Did I really mean that? Applying automatic summarisation techniques to formative feedback. In Galia Angelova, Kalina Bontcheva, and Ruslan Mitkov, editors, *RANLP*, pages 277–284. RANLP 2011 Organising Committee / ACL, 2013.

[13] John R. Firth. *A Synopsis of Linguistic Theory, 1930-1955*. 1957.

[14] Ilias Flaounas, Omar Ali, Thomas Lansdall-Welfare, Tijl De Bie, Nick Mosdell, Justin Lewis, and Nello Cristianini. Research methods in the age of digital journalism: Massive-scale automated analysis of news-content–topics, style and gender. *Digital Journalism*, 1(1):102–116, 2013.

[15] International Organization for Standardization. ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs) - Part 11 : Guidance on usability. Technical report, International Organization for Standardization, Geneva, 1998.

[16] Yoav Goldberg and Omer Levy. word2vec Explained: deriving Mikolov et al.'s negative-sampling word-embedding method. *CoRR*, abs/1402.3722, 2014.

[17] Peter O. Gray. *Psychology*. Worth Publishers, 2006.

[18] Johan Gunnarsson. Most viewed articles on Wikipedia 2013. `https://tools.wmflabs.org/wikitrends/2013.html`. [Online; accessed 25-May-2015].

[19] Zellig Harris. Distributional structure. *Word*, 10(23):146–162, 1954.

[20] Hendrik Heuer. Topic comparison tool. `https://github.com/h10r/topic_comparison_tool`, 2015.

[21] Minqing Hu and Bing Liu. Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, pages 168–177, New York, NY, USA, 2004. ACM.

[22] Matthew L. Jockers. *Macroanalysis: Digital Methods and Literary History*. Topics in the Digital Humanities. University of Illinois Press, 2013.

[23] Matthew L. Jockers and David Mimno. Where do themes occur in novels? `http://mimno.infosci.cornell.edu/novels/topics.json`. [Online; accessed 27-April-2015].

[24] Jussi Karlgren and Magnus Sahlgren. From Words to Understanding. Technical report.

[25] Jussi Karlgren, Magnus Sahlgren, Fredrik Olsson, Fredrik Espinoza, and Ola Hamfors. Usefulness of Sentiment Analysis. In Ricardo A. Baeza-Yates, Arjen P. de Vries, Hugo Zaragoza, Berkant Barla Cambazoglu, Vanessa Murdock, Ronny Lempel, and Fabrizio Silvestri, editors, *ECIR*, volume 7224 of *Lecture Notes in Computer Science*, pages 426–435. Springer, 2012.

[26] Samuel Kaski. Dimensionality reduction by random mapping: fast similarity computation for clustering. In *Neural Networks Proceedings, 1998. IEEE World Congress on Computational Intelligence. The 1998 IEEE International Joint Conference on*, volume 1, pages 413–418 vol.1, May 1998.

[27] Soo-Min Kim and Eduard Hovy. Determining the Sentiment of Opinions. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, Stroudsburg, PA, USA, 2004. Association for Computational Linguistics.

[28] David Lazer, Alex Pentland, Lada Adamic, Sinan Aral, Albert-László Barabási, Devon Brewer, Nicholas Christakis, Noshir Contractor, James Fowler, Myron Gutmann, Tony Jebara, Gary King, Michael Macy, Deb Roy, and Marshall Van Alstyne. Computational Social Science. 2009.

[29] Quoc V. Le and Tomas Mikolov. Distributed Representations of Sentences and Documents. *arXiv preprint arXiv:1405.4053*, 2014.

[30] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, May 2015.

[31] Omer Levy and Yoav Goldberg. Dependency-Based Word Embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 302–308, Baltimore, Maryland, 2014. Association for Computational Linguistics.

[32] Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *TACL*, 3:211–225, 2015.

[33] Annie Louis and Ani Nenkova. Automatically Assessing Machine Summary Content Without a Gold Standard. *Comput. Linguist.*, 39(2):267–300, June 2013.

[34] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP Natural Language Processing Toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, 2014.

[35] Richard E. Mayer, Sherry Fennell, Lindsay Farmer, and Julie Campbell. A Personalization Effect in Multimedia Learning: Students Learn Better When Words Are in Conversational Style Rather Than Formal Style. *Journal of Educational Psychology*, 96(2):389–395, June 2004.

[36] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient Estimation of Word Representations in Vector Space. *CoRR*, abs/1301.3781, 2013.

[37] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. word2vec. `https://code.google.com/p/word2vec/`, 2013. [Online; accessed 18-May-2015].

[38] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic Regularities in Continuous Space Word Representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT-2013)*. Association for Computational Linguistics, 2013. [Online; accessed 18-March-2015].

[39] George A. Miller. The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information. *The Psychological Review*, 63(2):81–97, March 1956.

[40] George A. Miller. WordNet: A Lexical Database for English. *Communications of the ACM*, 38:39–41, 1995.

[41] George A. Miller and Walter G. Charles. Contextual correlates of semantic similarity. *Language & Cognitive Processes*, 6(1):1–28, 1991.

[42] Jakob Nielsen. 10 usability heuristics for user interface design. *Fremont: Nielsen Norman Group.[Consult. 20 maio 2014]. Disponível na Internet*, 1995.

[43] O'Reilly. Formula - Head First - Series - O'Reilly Media. `http://shop.oreilly.com/category/series/head-first/formula.do`. [Online; accessed 28-April-2015].

[44] Bo Pang and Lillian Lee. A Sentimental Education: Sentiment Analysis Using Subjectivity. In *Proceedings of ACL*, pages 271–278, 2004.

[45] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Oliver Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, and Edouard Duchesnay. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[46] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. GloVe: Global Vectors for Word Representation. In *Proceedings of EMNLP*, 2014.

[47] Jeffrey Rubin and Dana Chisnell. *Handbook of Usability Testing: Howto Plan, Design, and Conduct Effective Tests*. Wiley Publishing, 2 edition, 2008.

[48] Magnus Sahlgren. An introduction to random indexing. In *Methods and applications of semantic indexing workshop at the 7th international conference on terminology and knowledge engineering, TKE*, volume 5, 2005.

[49] Magnus Sahlgren. *The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations Between Words in High-dimensional Vector Spaces*. SICS dissertation series. Department of Linguistics, Stockholm University, 2006.

[50] Hinrich Schütze. Word Space. In *Advances in Neural Information Processing Systems 5, [NIPS Conference]*, pages 895–902, San Francisco, CA, USA, 1993. Morgan Kaufmann Publishers Inc.

[51] Jonathon Shlens. A Tutorial on Principal Component Analysis. *CoRR*, abs/1404.1100, 2014.

[52] Richard Socher. *Recursive Deep Learning for Natural Language Processing and Computer Vision*. PhD thesis, Stanford University, 2014.

[53] Richard Socher, Yoshua Bengio, and Christopher D. Manning. Deep Learning for NLP (Without Magic). In *Tutorial Abstracts of ACL 2012*, ACL '12, pages 5–5, Stroudsburg, PA, USA, 2012. Association for Computational Linguistics.

[54] Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Stroudsburg, PA, October 2013. Association for Computational Linguistics.

[55] Stephen M. Stahl, Richard L. Davis, Dennis H. Kim, Nicole Gellings Lowe, Richard E. Jr Carlson, Karen Fountain, and Meghan M. Grady. Play it Again: The Master Psychopharmacology Program as an Example of Interval Learning in Bite-Sized Portions. *CNS Spectrums*, 15(08):491–504, 2010.

[56] Peter D. Turney. 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*, ACL '02, pages 417–424, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[57] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(2579-2605):85, 2008.

[58] Wikipedia. Floorball – Wikipedia, The Free Encyclopedia. `http://en.wikipedia.org/w/index.php?title=Floorball&oldid=659332393`, 2015. [Online; accessed 28-April-2015].

[59] Wiktionary. Frequency lists. `https://en.wiktionary.org/w/index.php?title=Wiktionary:Frequency_lists/PG/2005/10/1-1000&oldid=5621255`. [Online; accessed 31-March-2015].

[60] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing Contextual Polarity in Phrase-level Sentiment Analysis. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, HLT '05, pages 347–354, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[61] Ludwig Wittgenstein. *Philosophical Investigations*. Basil Blackwell, Oxford, 1953.

[62] Robert H. Wozniak. Introduction to Memory Hermann Ebbinghaus (1885/1913). `http://psychclassics.yorku.ca/Ebbinghaus/wozniak.htm#f2`, 1999. [Online; accessed 05-May-2015].

[63] Jeonghee Yi, Tetsuya Nasukawa, Razvan Bunescu, and Wayne Niblack. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. In *In IEEE Intl. Conf. on Data Mining (ICDM*, pages 427–434, 2003.

[64] Radim Řehůřek. Experiments on the English Wikipedia. `https://radimrehurek.com/gensim/wiki.html`. [Online; accessed 31-March-2015].

[65] Radim Řehůřek. Making sense of word2vec. `http://radimrehurek.com/2014/12/making-sense-of-word2vec/`, December 2014. [Online; accessed 27-April-2015].

[66] Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.