ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics

Master Thesis Data Science and Marketing Analytics

Comparison of Bag-Of-Words and Doc2Vec Sentence-Vector Representations in Graph-Based Summarization of Earnings Call Transcripts

Student: Sadykova, A.A. (423526)
Supervisor: Raviv, E.
Second assessor: Donkers, A.C.D.

Date final version: 21.07.2020

Abstract

This research illustrates the use of an automatic summarization algorithm in the domain of corporate disclosure. The goal of this study was to compare a graph-based automatic summarization algorithm that incorporates semantic inter-sentence similarities to one that only takes into account lexical similarities, by producing summaries of 20 earnings call transcripts. The two algorithms differ in their sentence-vector representations: the lexical summarization algorithm employs a bag-of-words (BOW) model to obtain sentence-vectors, while the semantic summarization algorithm makes use of sentence embeddings (Doc2Vec). After producing the summaries and evaluating them with a help of six human judges, it has been concluded that the semantic algorithm produces summaries containing information that is more useful for making investment decisions about a company.

Contents

Chapter 1

Introduction

The trend of growing volume and complexity of corporate disclosure (Dyer et al., 2016) has the benefits of reducing information asymmetry and alleviating agency problems in capital markets (Healy & Palepu, 2001). However, more is not always better. Given people's limited processing power and a narrow time frame within which the information is relevant, users of corporate disclosure (i.e. investors, asset managers, financial analysts and others) are at risk of being subject to information overload (Kågebäck et al., 2014; Paredes, 2003). The inefficient processing of information, in turn, may result in impaired decision making among market participants, counterbalancing the benefits of the more extensive corporate disclosure (Paredes, 2003).

One example of a particularly lengthy type of corporate disclosure is *earnings calls*. An earnings call is a type of voluntary disclosure conducted in a form of a webcast conference call (Hollander et al., 2010). Earnings calls provide investors and market professionals with additional information about the firm's current and future performance, complementary to the mandatory disclosure and news releases (Bowen et al., 2002; Frankel et al., 1999). Moreover, these calls are transcribed and are provided as text documents to the public.

Since earnings call contents are available in text format, the information overload problem can be addressed with the help of a text mining technique called *automatic text summarization*. Automatic text summarization is defined as the process of using software to produce a summary - a compressed form of a document which contains

its most relevant information as well as the main idea, while being shorter than the original document (Allahyari et al., 2017; Gupta & Lehal, 2009). Availability of a high-quality summary gives the user a preview of the document contents and can aid them in judging whether it is worth reading entirely. However, producing a manual summary can be time-consuming and highly dependent on the judgement of the person involved. Therefore, an automated summarization tool can aid humans in processing lengthy and complex texts at greater speeds and with less bias (Gupta & Lehal, 2009).

Even though automatic text summarization can benefit users of text in different domains, it can particularly prove valuable in aiding investors and market professionals with investment decisions, due to the importance of having quick access to up-to-date information. An automatically produced summary can be useful to earnings call participants in providing an unbiased recap of the main points discussed, or to individuals who could not follow the call and therefore need a quick overview of its contents, in order to decide on whether to proceed with reading the full transcript.

A common approach to automatic summarization is sentence extraction, which presents the most relevant information from a document by selecting a few most important sentences (Nenkova et al., 2006). There is a multitude of ways in which important sentences can be identified. One common group of sentence extraction approaches are graph-based summarization algorithms, which this study will focus on. Graph-based approaches treat the input document as a set of interrelated units, i.e. sentences, and use those relations to determine the most central sentences that will be included in the summary.

One important step in the process of graph-based summarization is establishing inter-sentence links by measuring their similarities, since this information is then utilized in ranking the sentences. However, the most influential graph-based approaches that were presented in prior literature (Erkan & Radev, 2004; Mihalcea & Tarau, 2004) only take into account lexical properties of the sentences, overlooking their semantic aspects when inferring similarities of sentences (Allahyari et al., 2017). Lexical similarity between pieces of text is defined as similarity based on the common words they contain, while semantic similarity is based on their meaning. Ignoring semantic similarity may lead to inferior summarization results in the case of earnings call transcripts, due to the specifics of the financial domain language and the speech format of the text. For instance, earnings calls participants use a lot of abbreviations and their full forms interchangeably: a lexical similarity measure would underestimate the similarity between two sentences that talk about “deferred tax assets” and

"DTA". Incorporating the semantics may result in more useful summaries of earnings call transcripts by selecting sentences that are central in terms of their meaning, rather than their lexical properties. Therefore, the goal of this study is the following:

To examine whether a graph-based summarization algorithm that incorporates semantic inter-sentence similarities, as opposed to lexical similarities, produces more useful summaries of earnings call transcripts.

The contribution of this study is therefore twofold: it illustrates the application of automatic text summarization in the domain of corporate disclosure (1), and attempts at demonstrating the potential benefits of incorporating semantic relationships into a graph-based summarization algorithm (2).

Further discussion will now unfold by first discussing the supporting literature in the areas of information overload, text summarization and sentence-vectors representation. After that, the dataset containing documents to be summarized will be introduced and corresponding pre-processing will be described. After that, methodology relating to the process of text summarization will be explained. Namely, two graph-based summarization algorithms, which differ in their sentence-vector representation, will be implemented and compared in this study. That is, a bag-of-words model will be applied to produce sentence-vectors used for obtaining lexical similarities (Erkan & Radev, 2004), while semantic similarities will be derived by using Doc2Vec sentence embeddings as an input (Mikolov & Le, 2014). This thesis will then be wrapped up by discussing the results, addressing the research goal, pointing out the limitations of the study from which future research suggestions will ensue.

Chapter 2

Related Literature

In order to lay the foundation for this study, this section will start off by discussing the previous literature that reveals the need of building an automatic summarization tool in the setting of earnings call transcripts, by presenting theoretical and empirical justifications. Next, some background on automatic text summarization will be provided, consequently narrowing it down to discussing some notable work on graph-based summarization, concluding with influential papers that tackle the problem of sentence-vector representation.

2.1 The Need for Automatic Text Summarization in Corporate Disclosure Domain

As emphasized in the introduction, the increase in corporate disclosure has an effect of subjecting its users, i.e. financial market participants, to information overload. Information overload among the users of corporate disclosure has been acknowledged in a number of papers. For instance, Paredes (2003) challenges the ongoing tendency of regulators to require more corporate disclosure with the aim to reduce information asymmetry between public companies and market participants. The author considers the perspective of the users of this information by arguing that the market participants' rationality is limited, as opposed to what the efficient markets hypothesis assumes (Fama, 1970). He indicates that as the volume and complex-

ity of information presented to an individual increases, the cognitive effort they put into processing this information decreases. This phenomenon is called information overload. Information overload may cause an investor to employ simplified decision making strategies such as certain heuristics, which in turn may lead to a suboptimal investment decision. The author concludes with some suggestions calling the regulators to scale back or simplify the requirements for corporate disclosure.

Miller (2010) supports the above with empirical findings. The author studied the impact of disclosure length on the trading volumes of corresponding stock by examining 10-K reports and corresponding trading volumes around the date of their release. In this paper, the length of the reports is measured as the number of words they contain. The effect on the trading volume was studied by controlling for other factors that affect trading volumes such as profitability or the date of a prior earnings announcement. It was discovered that the lengthier 10-K filings result in less trading, especially among individual investors, which supports the notion that it is more costly for individual investors to efficiently process overflowing information.

In both of the above studies, the authors agree that the implications of information overload can be mitigated by loosening the regulations, by the use of other sources of information such as news or analysts insights, as well as investors themselves using tools available due to technological advances in order to decrease the costs of processing large volumes of information.

Automatic text summarization has been extensively used to address the problem of information overload in different domains, such as news websites (Lee et al., 2005), scientific articles (Luhn, 1958), biomedical documents (Plaza et al., 2011) etc. The current study aims to contribute to addressing the problem of information overload among the users of corporate disclosure, since there appears to be a gap in the literature that would demonstrate the application of automatic text summarization in this domain.

2.2 Graph-based Automatic Text Summarization

One of the early attempts at automatic text summarization has been made by Luhn (1958), who introduced an algorithm that performs summarization of scientific articles. This was done by extracting highly important sentences from a text. Those

sentences were detected by first obtaining the frequency of every term in the article, thus identifying the most important (i.e. frequent) terms, the presence of which would in turn increase the significance of a sentence that contains them. The sentences with the highest importance score were more likely to be included in a summary.

Since then, a multitude of summarization approaches have been developed, which can be classified based on several criteria. For example, with respect to the way the output is presented, automatically created summaries can be *extractive* and *abstractive*. In the former approach, the key sentences from a text are included into a summary in the exact form they appear in the original text, while the latter approach presents the main idea of the text by paraphrasing the information from the original text. Abstractive summarization approach requires using advanced natural processing techniques and often builds up on extractive summarization, since the most important information needs to be identified first. This research thus focuses on producing extractive summaries.

Another dimension along which (extractive) automatic summarization techniques can differ is the underlying summarization algorithm. Those can be heuristics-based (Edmundson, 1969; Luhn, 1958), topic-based (Celikyilmaz & Hakkani-Tur, 2010; Steinberger & Jezek, 2004), based on machine-learning (Wang et al., 2008) as well as graph-based (Erkan & Radev, 2004; Mihalcea & Tarau, 2004). The basic text summarization techniques assign sentence importance based on certain heuristics such as appearance of high frequency words and phrases, sentence length, sentence position and others. The topic-based approaches aim to identify the most important topic(s) in the text and then select the sentences that best represent those topics. In the presence of high quality training data, the task of determining important sentences can be tackled with the help of supervised machine learning techniques, by using various sentence-specific features in order to predict whether a given sentence is to be included in the summary (Gambhir & Gupta, 2017). This study will focus on graph-based summarization, which is more practical in the case of earnings calls transcripts, due to unavailability of unbiased reference summaries that could be used for training and updating machine learning models. Graph-based methods are characterized by representing the document as a graph, a mathematical structure that consists of nodes and edges. The nodes correspond to the pieces of text that need to be extracted (such as sentences or paragraphs), while the edges represent pairwise connections between them (i.e. similarities). The idea is to apply a ranking algorithm which would assign importance scores to the nodes, so that they can be

ranked and thus the most important nodes, that is, sentences or paragraphs can be determined and included in a summary.

One of the first papers in which a graph-based approach was applied in the context of text summarization was by Salton et al. (1997). In this paper, the authors introduced an algorithm that summarizes long texts by extracting salient paragraphs. The underlying ranking algorithm is based on degree centrality: nodes corresponding to the paragraphs as well as the edges between them are presented in a graph, after which a certain threshold for the minimum similarity is selected. All connections that are below this threshold are eliminated, and for each node (paragraph), the number of remaining edges that are connected to it (that is, similar paragraphs) are counted. A paragraph which has more paragraphs that are similar to it is then considered more important and is more likely to be included in the summary. The method was tested by summarizing encyclopaedia articles and comparing them to human-generated summaries by computing paragraph overlap. The method performed considerably better than the baseline of selecting random paragraphs.

A few years later, Mihalcea and Tarau (2004) and Erkan and Radev (2004) independently introduced similar ranking algorithms, TextRank and LexRank respectively, which added an important modification to the algorithm of Salton et al. (1997). In Salton et al. (1997) importance of a node depends on the number of nodes it connects to, while TextRank and LexRank take a step further, by having importance of a node also dependent on the importance of the nodes it is connected to. Both works take inspiration from Google's PageRank algorithm (Brin & Page, 1998), which is used to determine the prestige of a webpage given a query, the difference being that TextRank and LexRank are used to rank pieces of text in a document. However, the above methods fall short in an important aspect of the summarization process - sentence-vector representation. In these works, the authors use text representations that employ the bag-of-words assumption: each word is treated as having its own dimension and being uncorrelated with other words. This has an effect of the similarities between the sentences being lexical, and not taking semantic relationships (i.e. meaning) into account. The authors themselves note that the computation of similarity as part of the summarization algorithms can be improved by incorporating more advanced sentence representations. This shortcoming can therefore be solved by altering the sentence-vector representation so that the semantics of the input text is captured.

2.3 Semantic Representation of Text

One of the notable works that introduced a widely used approach that models semantic meaning of words was by Mikolov et al. (2013). The output of this approach are continuous word-vectors (word embeddings), which capture contextual meaning of the words as well as certain relationships and analogies between them. The main idea is that the meaning of a word is determined by its context, i.e. surrounding words. The authors proposed two methods which are based on artificial neural networks, prediction models which can capture complex non-linear relationships in the data (Mitchell, 1997). One of them, Continuous bag-of-words (CBOW), obtains word-vectors by training a neural network on predicting each word in the text from surrounding words, while in the second, Skip-gram model, the surrounding words are predicted from a word of interest. In both cases, the word-vectors are trained using a neural network model with a linear hidden layer. This is more computationally efficient than the earliest neural language model for obtaining word embeddings (Bengio et al., 2003), in which the authors used a nonlinear hidden layer. Nevertheless, it is still able to capture more complex word relationships than linear approaches for obtaining continuous word-vectors, such as Latent Semantic Indexing (Dumais et al., 1988). The authors tested the CBOW and Skip-gram approaches by applying the obtained word-vectors on a word similarity task, and the two methods have outperformed the other methods for obtaining word-vectors.

However, the automatic text summarization task requires text representation to be on a sentence level. Therefore, a method which can similarly capture the meaning of longer pieces of texts such as sentences is needed. Mikolov and Le (2014) have identified that the context of a word can consist not only of other words, but also defined by the phrase, sentence, or a paragraph it is a part of. Building upon this intuition, they introduced Doc2Vec, a method for obtaining vector representations for longer sequences of words, including sentences. This method is simply an extension of the CBOW and Skip-gram approaches introduced in their previous work, except that the IDs of a sentence the word is part of are treated as extra words. In this way, corresponding sentence-vectors (sentence embeddings) can be obtained, which can be used to model various relationships between sentences, including semantic similarities which are needed for automatic text summarization. This approach was tested on the task of text classification in the context of sentiment analysis, as well as information retrieval. In both tasks, the sentence embeddings proved to perform better than different benchmarks such as the bag-of-words representation of sentences as well as simply averaging word embeddings produced by CBOW and Skip-gram.

The current study, in turn, contributes to the literature by examining performance of the Doc2Vec sentence-vectors on the task of automatic summarization, comparing it to the traditionally used bag-of-words representation.

Chapter 3

Data

The data used to answer the research question was extracted from Bloomberg Transcript, and consists of 20 transcripts of earnings calls conducted in 2016-2017 by different public companies. Each transcript corresponds to a separate PDF file. Even though the documents come in a text format, their structure is standardized. Each page of the PDF documents contains a page number, and a page header with the quarter-end financial results that is present across all pages. The main text starts with a document heading that states the reporting period the call is dedicated to, followed by lists of company participants and external participants (under the headings 'Company Participants' and 'Other Participants' respectively). This is followed by the text corresponding to the Management Discussion Section, where the company participants and an operator take turns to speak. Each utterance starts with a heading that states who is speaking. This is then followed by a transcript of the Q&A session (under the 'Q&A' heading), where external participants ask questions and company participants respond, and an operator curates the discussion. Each document is concluded by a standardized disclaimer statement and a copyright notice. Figure 3.1 provides examples of the first, last pages and the page where the Q&A section begins, taken from one of the transcripts. This research focuses on summarizing the *Management Discussion Section* and leaves out the Q&A section, since its question-answer structure would require a different approach to summarization.

Figure 3.1: Sample pages from one of the earnings call transcripts.

In order to prepare the text for automatic summarization, first, some cleaning was performed on each document using regular expressions - special string combinations that are used to match certain patterns in a text. The cleaning process involved removing page headers and page numbers, after which the text corresponding to all pages in a document was merged into one continuous string. After that, the company participants list was extracted into a separate vector, which was used to remove the headings that announce the speakers in the Management Discussion Section (all of them are company participants). The string pattern used to match these headings is preceded and followed by a line break ('$\backslash r \backslash n$'), allowing to avoid removing in-text mentions of company participants. After that, the document heading, the list of other participants, the Q&A section, the disclaimer statement, and the copyright notice were removed. Moreover, all utterances corresponding to the operator were removed, since these pieces of text do not convey any useful information and serve as a transition between the speeches of company participants. The above steps resulted in 20 documents, each containing continuous text representing the sentences from the Management Discussion Section.

Once the redundant parts of the documents were removed, additional preprocessing had to be performed in order to transform the text into a more structured form which would be readable and analyzable by an algorithm. First, in order to keep track of the data, each document was assigned a numeric ID (from 1 to 20). Each document was further split into sentences, since the summarization is performed by analyzing the data on a sentence level. This was done using a set of rules, like defining the

boundaries of a sentence as '.' , '?', '!' which are followed by a capital letter, as well as taking into account exceptions such as 'Mr.', 'Mrs.' etc. Each sentence, in turn, was assigned a numeric ID (from 1 to the length of the document, defined as the number of sentences).

Further preprocessing involved making choices with regards to several aspects, some of which diverged depending on the sentence-vector representation applied, since the input text requirements can differ across the two algorithms utilized in this study. The choices needed to be made with regards to the following preprocessing steps: first, whether or not all characters should be set to lowercase, removing (certain) punctuation, extra white space, how numbers should be treated. Another important decision is about whether or not stop words should be removed. Those are the highly common words in the language that do not provide much additional meaning on their own (e.g. 'a', 'the', 'he', 'she' etc.). Finally, a decision about performing stemming on the remaining words is to be made, which means using a set of rules to transform a word into its base (root) form. In this way, a set of highly related words which only differ in their form can be treated as the same item. These steps will be addressed in the methodology section (Section 4.1.3), once the two sentence-vector representation approaches are explained and it becomes clear what the requirements for the input text are.

Chapter 4

Methodology

As was stated earlier, this research focuses on the graph-based approach to summarization. Graph-based summarization is performed by determining pairwise inter-sentence links in a document, which can be visualized in a graph. The resulting graph consists of nodes, which represent sentences, and those are interlinked by edges, which represent the extent to which the sentences are similar. The rationale behind this approach is that the sentences that are strongly linked to a lot of other sentences capture the main idea of the document well and are therefore assigned higher importance (Mihalcea & Tarau, 2004). Moreover, the importance of a sentence also depends on the importance of other sentences that are linked to it. Taking into account these factors, an algorithm can be applied in order to rank the sentences in the document of interest, afterwards extracting the most important ones to be included in the summary.

There are three important steps that need to be performed to obtain a summary using a graph-based approach (Figure 4.1). First, a sentence-vector representation is obtained (1). After that, a graph that takes into account pairwise similarities between the sentence-vectors is constructed (2). Then, a ranking algorithm is applied in order to determine how important each sentence is (3). Using this ranking, the top K sentences are then selected to be included in the summary, which are then sorted by sentence ID, so that they appear in the original order.

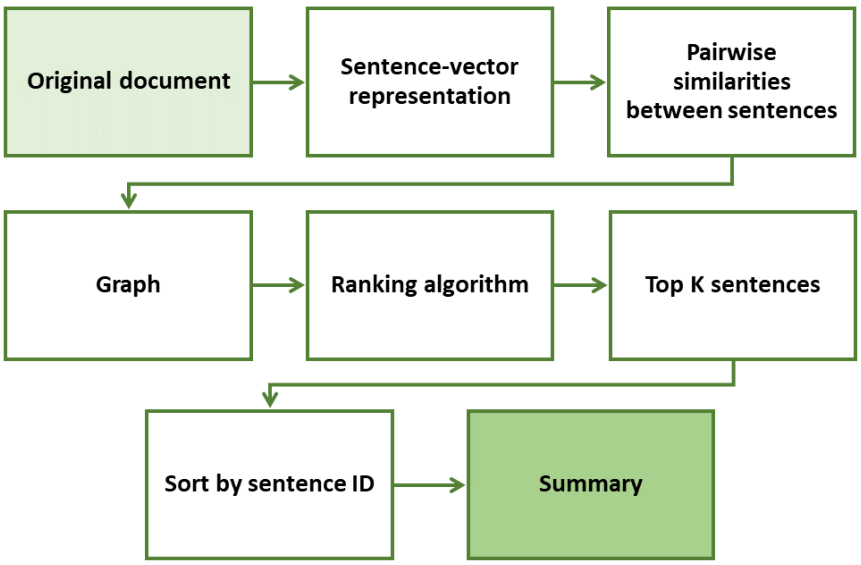


Figure 4.1: Graph-based summarization process.

In this study, the implications of changing sentence-vector representation were examined, keeping other aspects unchanged. The next subsection (Section 4.1) will describe the two different sentence-vector representations that were compared: bag-of-words model, which captures lexical properties of the sentences, and Distributed Memory model (Doc2Vec), which takes into account their semantic properties. Section 4.2 will then dive into the process of computing pairwise similarities between sentences using the two different sentence-vector representations, as well as constructing a corresponding graph. Finally, the algorithm used to determine the importance of the sentences from the information obtained in the first two steps will be explained in Section 4.3.

The above steps will result in two different summaries for each earnings call transcript. In order to assess the effect of altering the sentence-vector representation when performing transcript summarization, this research will benefit from having an evaluation procedure in place, which will be introduced in Section 4.4.

4.1 Sentence-Vector Representation

As outlined above, the first step in the summarization process is obtaining a sentence-vector representation. This section will describe the two approaches that will be compared in this study, namely, the bag-of-words (BOW) model with TF-IDF weighting, which captures the lexical properties of the sentences, and Doc2Vec sentence embeddings, which represent the semantics of the sentences.

4.1.1 Bag-of-words (BOW) Representation

Bag-of-words (BOW) is a traditional way to model text, which represents sentences as fixed-length vectors. In this approach, a document-term matrix (DTM) is constructed, where each row corresponds to a sentence and every column represents a separate term from the document vocabulary, that is, a set of unique terms corresponding to the document. Each value in the matrix corresponds to the *Term Frequency (TF)* weighted by the *Inverse Document Frequency (IDF)*. TF shows how many times a term occurs in a sentence. However, if a certain term is present in most sentences in the document, it would not be so valuable to this particular sentence. Therefore, a measure of relative importance of a term in a sentence should be considered. This is achieved by multiplying the TF by the term's IDF score, which is calculated as:

$$IDF_{t,D} = \log(\frac{N}{n_t}), \tag{4.1}$$

where N is the total number of sentences, and n_t is the number of sentences in which term t is present. Multiplying the term occurrence by its IDF score has an effect of downweighing the terms that are common in the document in general, and assigns higher importance to the terms that are more frequent in the sentence of interest, but are not common in the rest of the document. To sum up, the underlying sentence vector representation has a form of a matrix, with each entry (s, t) corresponding to the TF-IDF score of term t in sentence s.

4.1.2 Doc2Vec Sentence Embeddings Representation

In the BOW approach, each term has its own dimension, and therefore it is assumed that all terms are unrelated to each other. For instance, words 'company' and 'firm' will be considered as different as 'company' and 'cat' are, even though the former word pair are closer to each other in terms of their meaning. That is, the bag-of-words assumption does not take the semantics behind the word tokens into account. This may pose a problem when looking at the similarities between sentences where certain word relationships, such as synonyms and antonyms, are involved. There are other disadvantages of the BOW model, such as not taking the word order into account ('this is surprising' will be equal to 'is this surprising'), and high dimensionality, which increases computational complexity.

One solution to this problem can be to use continuous vector representation of sentences, i.e. sentence embeddings. A sentence embedding is a mapping of sentences into a vector space with a predefined number of dimensions, where each real-number vector captures the meaning of the corresponding sentence. Semantically similar sentences will be mapped to embedding vectors that are relatively close to each other in the vector space.

A popular way to embed a sentence has been introduced by Mikolov and Le (2014) and is known as *Doc2Vec* algorithm. This algorithm is an extension of *Word2Vec*, a comparable algorithm for obtaining embeddings for words, which incorporate their meaning. Both algorithms are based on the *Distributional Hypothesis* (Harris, 1954), according to which the meaning of a word is defined by its context, and hence, similar words appear in similar contexts. In the Doc2Vec algorithm, this idea is extended to longer sequences of text, such as sentences, paragraphs, or even entire documents.

Doc2Vec sentence embeddings can be obtained in two ways: from the *Distributed Memory model (DM)* or the *Distributed bag-of-words model (DBOW)*. The difference between the two is that the latter does not take the word order into account, which is crucial to consider when working with sentences. Therefore, it is decided to use the DM model to obtain the sentence representations in this study.

The Distributed Memory variation of Doc2Vec is an extension of the Continuous bag-of-words (CBOW) variation of Word2Vec (Mikolov et al., 2013). Therefore, it is reasonable to first describe how word embeddings using CBOW are obtained.

Feedforward Neural Network

In the CBOW approach, the word embeddings are a by-product of a prediction task fulfilled by a *feedforward neural network* model. A feedforward neural network is a prediction model characterized by a network of interconnected units, called neurons. It is trained to approximate complex non-linear relationships in the data via a process that imitates a human brain. The neurons are organized in layers: the *input layer*, where each neuron represents a predictor feature, the *output layer*, which produces the output vector the network is trained to predict, as well as *hidden layers*, where the relationships are captured. The neurons of the hidden and output layers receive the weighted sum of the inputs coming in from the neurons of a preceding layer, and transform the result through an activation function, applied to produce the output. The result is then an input of the next layer, if present, or the final output of the model.

A neural network is trained to obtain the weights matrices which hand over the values from one layer to the next. This is done via the *backpropagation algorithm*, which involves several steps. First, weight matrices with random values are initialized (1). After that, each training instance is fed forward through the network by computing the weighted sum using the previously initialized weights, and applying a corresponding activation function at each layer, eventually producing the response values at the output layer (2). Next, the *error* is calculated using a certain *loss function* which compares the calculated response value against the actual response value (3). The error is then propagated back using the same weights (4), after which the gradient is obtained, i.e. the partial derivatives with respect to every weight in the network (5). This results in a vector that points in the direction in which the weights in the matrices are then updated (6), i.e. the direction of minimizing the loss. The weights are updated after every training instance is propagated forward and back. Passing the entire set of training instances through the model adds up to an *epoch* (7). Stages 2-7 are repeated for a number of epochs until the convergence is achieved, i.e. when the loss reduction becomes negligible. Such iterative method for finding the minimum of the loss function is called *stochastic gradient descent*.

Each time the weights are updated, the update step size is scaled down by multiplying it by a *learning rate* which ranges between 0 and 1, allowing to avoid converging at suboptimal weight values. However, setting a learning rate that is too low may result in long training times. Therefore, it is common to apply a higher learning rate at the start of the training process and decrease it with every epoch so that update steps

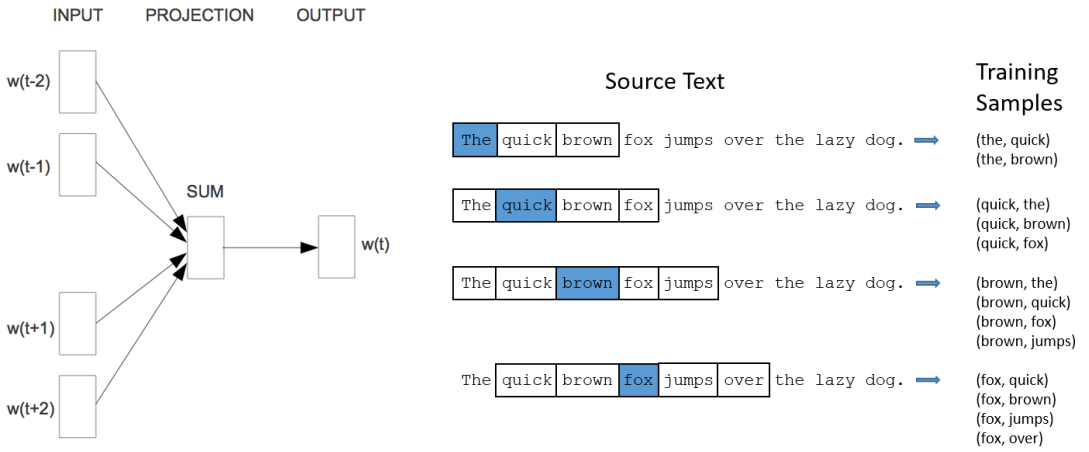
are smaller at the end, in order to not overshoot the minimum.

Continuous Bag of Words Model for obtaining word embeddings

A neural network in the setting of CBOW is trained on the task of predicting the target word from the words that surround it, i.e. its context. However, the neural network is not trained for prediction purposes. The final goal is to obtain the weight matrix between the input and the hidden layers, which contains the word embeddings. By training the neural network on predicting a word from its context, the resulting weight matrix with word embeddings captures the contextual meaning of the words as well as the similarity and relationships between them (Mikolov et al., 2013).

The process of obtaining the word embedding vectors involves the following steps: first, the training instances are extracted from the entire corpus given a predefined window size, which represents the context of a target word (1), after which a neural network is trained on these instances (2).

The context of a word depends on a chosen window size: the number of preceding and subsequent words around the word of interest. Schematically, the CBOW model is visualized in Figure 4.2a: the target word to be predicted is denoted as $w(t)$, and $w(t+j)$ are the context words that act as predictors, where $j = -2, -1, 1, 2$ denotes how far the context words are from the target word, where the window size is equal to 2 in this example. Given a window size, the construction of the training instances is illustrated in Figure 4.2b. It can be seen that each word in the sequence at some point takes the role of a target word, which forms the output value for the prediction task, while the context words form the input values of the neural network.



(a) Structure of CBOW model (b) An example of constructing training instances

Figure 4.2: The Continuous bag-of-words (CBOW) model

Figure 4.3 illustrates the architecture of the neural network utilized in the CBOW model, which applies a slight modification to the feedforward neural network described in the previous subsection. The input layer of the neural network is represented by C one-hot encoded row vectors of dimension $1 \times V$, where C is the number of context words and V is the vocabulary size of the corpus. In every vector, 1 corresponds to the respective context word and 0 to the rest of the terms - the neurons introduced in the previous subsection. Each of these vectors is multiplied by the weight matrix $\mathbf{W}$ of dimension $V \times N$. This has an effect of extracting the rows of the matrix $\mathbf{W}$ corresponding to the position of the context words in the vocabulary, resulting in C vectors of dimension $1 \times N$. Each of these vectors represents unique embeddings of the corresponding input words. The hidden layer of the neural network is a $1 \times N$ vector which is obtained by averaging the C embedding vectors. The activation function of the hidden layer is an identity function, that is, the values of the $1 \times N$ vector are also the output of the layer. The values produced by the hidden layer are then propagated to the output layer by multiplying the vector by matrix $\mathbf{W_1}$ of dimension $N \times V$, which contains the second set of weights in the neural network to be learnt. The dimension of the output layer is also equal to the size of the vocabulary $(1 \times V)$. The resulting vector is fed through an activation function, in this case *softmax*, which is used for multiclass classification. Softmax transforms the values in the vector into a probability distribution, such that each output neuron produces the probability that the term this neuron corresponds to is the word for

which the input term is a context word.

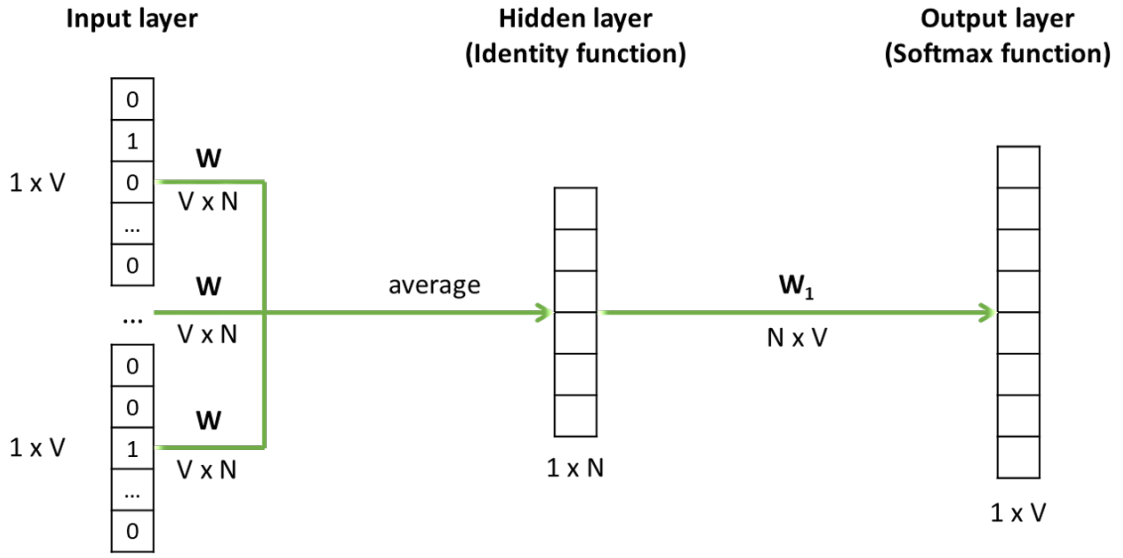


Figure 4.3: Architecture of the Continuous bag-of-words model.

Before defining the softmax function in the setting of the CBOW model, it is important to introduce the loss function, which is minimized with respect to the weights in the matrices $\mathbf{W}$ and $\mathbf{W_1}$, and is defined as the *negative log-likelihood*:

$$L = -\frac{1}{T}\sum_{t=j}^{T-j} \log p(w_t|w_{t-j}, ..., w_{t+j}), \tag{4.2}$$

where T is the length of the corpus (that is, the total number of words in the corpus), j is the window size, and t is the position of the target word of interest in the vocabulary. The corresponding probability is defined by the softmax function:

$$p(w_t|w_{t-j}, ..., w_{t+j}) = \frac{exp(y_{w_t})}{\sum_{i=1}^{V} exp(y_i)}, \tag{4.3}$$

where y_{w_t} is the pre-activation value of the output neuron at the position of the target word w_t, while y_i are the pre-activation values of the neurons corresponding to every

term in the vocabulary. The neural network is trained to minimize the negative log-likelihood, or equivalently, maximize the probability that the neural network predicts the actual target word, given the context words, averaged over all the words in the document that take on the role of the target word. This is done by applying stochastic gradient descent and backpropagation algorithm outlined before, to obtain the weights in matrices $\mathbf{W}$ and $\mathbf{W_1}$. The main goal is to obtain the matrix $\mathbf{W}$, its rows representing embedding vectors corresponding to each term in the vocabulary.

From Word2Vec to Doc2Vec

The *Distributed Memory (DM) model* is an extension of the CBOW model. The main idea of the DM model is that a sentence itself can act as a predictor of the target word, that is, be part of the context. By treating sentences as part of the context, it is possible to map the sentences to their unique embedding vectors in the same way it is done for words. This section will now point out the differences from the CBOW model in terms of the two main aspects: defining the context (1) and, as an implication, the architecture of the underlying neural network model (2).

As explained above, the sentence embeddings can be obtained by regarding a sentence as part of the context. This is done by adding the unique ID of every sentence in the corpus to the vocabulary, that is, treating them as additional terms. In order to capture the word order, the DM model defines the context of a target word as C preceding words plus the ID of the sentence the target word belongs to. Thus, for all target words in the corpus, $C + 1$ training instances will be formed for each. This means that in the DM model, the neural network is trained to predict the next word in the sequence, given the previous words and the sentence ID. The sentence ID token reflects the information missing from the rest of the context and can appear in the context of multiple target words, depending on the length of the corresponding sentence.

The above modification requires a slight adjustment to the neural network architecture, which is illustrated in Figure 4.4. The input layer now consists of C one-hot vectors of dimension $1 \times V$ for the words preceding a target word, and one one-hot vector of dimension $1 \times S$ (S is the number of sentences in the corpus), representing the sentence corresponding to the target word. Moreover, an additional matrix of weights $\mathbf{W_2}$ of dimension $S \times N$ is introduced that propagates the values in the sentence input vector to the next layer, alongside the $\mathbf{W}$ matrix that contains word

embeddings. Similar to the CBOW model, the word input vectors are multiplied by matrix $\mathbf{W}$, and the sentence input vector is multiplied by matrix $\mathbf{W_2}$, resulting in extracting $C+1$ embedding vectors of dimension $1 \times N$ (N is the predefined size of the embedding vector), the one extracted from $\mathbf{W_2}$ corresponding to the embedding vector of the sentence of interest. Then, an average of these vectors is taken to arrive at the hidden layer vector of dimension $1 \times N$.

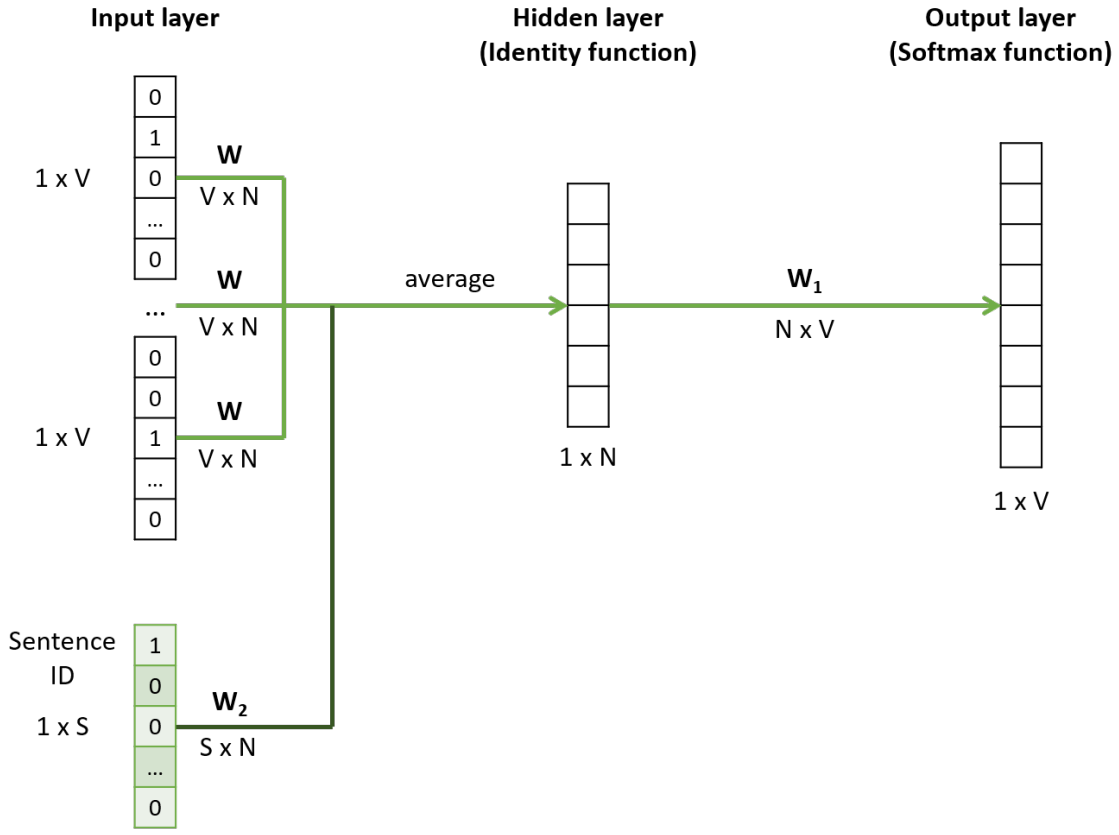


Figure 4.4: Architecture of the Distributed Memory model (Doc2Vec).

The next steps are the same as in the CBOW model: the values are propagated forward to obtain the prediction for the next word given the context, the loss is calculated by comparing the prediction with the actual target word, after which the backpropagation and stochastic gradient descent are applied iteratively to obtain the weights in matrices $\mathbf{W}$, $\mathbf{W_1}$ and $\mathbf{W_2}$. While each row of $\mathbf{W}$ corresponds to a unique embedding vector for each vocabulary term, the rows of $\mathbf{W_2}$ correspond to

embedding vectors for each sentence in the corpus. Just like with word embeddings, semantically similar sentences are mapped to embedding vectors that are relatively close to each other in the vector space.

In the Doc2Vec model, there are hyperparameters that need to be tuned in order to obtain optimal sentence representations, namely window size C and the embedding vector dimension N. The authors of Doc2Vec tune these values by evaluating the results on a downstream supervised task which uses the sentence embeddings as input, such as document classification. Hence, a labelled data set is needed in order to perform hyperparameter tuning in an automated manner. However, in this research, the sentence embeddings are used as an input to an unsupervised task - text summarization, for which no labelled data is available (as will be outlined in Section 4.4, the evaluation of the summaries has been conducted manually). Due to inability of checking the embeddings performance on the task of interest for tuning the hyperparameters, the general consistency of the obtained Doc2Vec model has been considered using a method outlined in gensim documentation, a Python library which offers tools for implementing Doc2Vec (Řehůřek, 2019). In this method, a new embedding vector is inferred for each sentence using the already trained model, as if they were new unseen sentences. This is done by fixing the weights of the word embeddings matrix $\mathbf{W}$, and expanding the sentence embeddings matrix with the "new" sentence, after which the weights are updated using backpropagation and gradient descent to obtain the embedding vector for this sentence. After that, the similarity between the inferred vector for the given sentence and all the pre-trained sentence embeddings are calculated, after which the rank of the original sentence is returned based on (self-) similarity. Ideally, a consistent model will rank the sentence as most similar to itself (rank for the sentence itself will be equal to one). This procedure has been performed on all sentences in the corpus and the proportion of the ranks equal to one has been calculated, that is, the proportion of the documents that are classified as most similar to themselves by the model. This proportion has been calculated by obtaining the Doc2Vec models using different configurations of C and N, and the combination that corresponds to the highest proportion has been selected as optimal.

4.1.3 Preprocessing Steps, Revisited

Now that the two methods for obtaining sentence-vector representations are explained, it becomes more clear what kind of input text they require. Revisiting the

steps described in Section 3, the two algorithms agree on calling for the following preprocessing steps. First, in both cases all characters were set to lowercase, since in general it is not necessary to differentiate between a capitalized word and its non-capitalized counterpart (e.g. 'company' and 'Company'). Second, earnings calls transcripts contain a lot of numbers (e.g. dollar amounts) all of which were replaced by a '123' string. This was done because the particular amounts do not convey meaningful information (e.g. neither of the algorithms can see that '$3' is greater than '$1'), but the fact that a number is present in a sentence may add to its meaning in this particular domain (e.g. two sentences that mention dollar amounts are similar to each other). Moreover, stemming was performed in both cases. In bag-of-words, it reduces the vocabulary size and prevents assigning highly similar words to separate dimensions (e.g. it is better if 'work', 'worked' and 'working' are reduced to 'work'). In Doc2Vec, stemming was used in order to shrink the vocabulary size and therefore reduce computation time.

The first preprocessing step with respect to which the two methods differ is punctuation removal. In bag-of-words, all punctuation was removed since it does not convey any additional meaning when treated as separate tokens. On the other hand, certain punctuation marks can be useful in defining the context, such as a question mark denoting a question, quotation marks conveying sarcasm, or an exclamation mark denoting certain emotions (dismay, excitement). Therefore, these three punctuation symbols were not removed in Doc2Vec. A similar point applies to stop words, since they do not convey much meaning when treated as separate tokens. Therefore, they were removed as a preprocessing step for the bag-of-words model. On the contrary, those words were not removed in Doc2Vec, since they can add to the context, for example, by helping to differentiate between meanings of some words ('to run' vs 'a run').

4.2 Constructing the Graph

Once the sentence-vector representations are obtained, the next step in the graph-based text summarization approach is to represent the document as a graph. Such a graph requires an $S \times S$ similarity matrix as an input, containing pairwise similarities between sentences. To calculate each entry in the matrix, a widely used *cosine similarity* metric has been applied, which measures the proximity between two non-zero sentence-vectors using the following formula:

$$\cos(\boldsymbol{a}, \boldsymbol{b}) = \frac{\sum_{n=1}^{N} a_n b_n}{\sqrt{\sum_{n=1}^{N} a_n^2}\sqrt{\sum_{n=1}^{N} b_n^2}}, \tag{4.4}$$

where **a** and **b** are two sentence-vectors of length N. Cosine similarity values lie in the range $[-1, 1]$, with -1 meaning that the sentences have an exact opposite meaning, 0 suggesting that they are unrelated, and 1 indicating that the sentences are exactly the same.

The elements of the resulting similarity matrix show the strength of the connections between sentences. In the case where the bag-of-words model is used for sentence representation those connections are lexical, while the Doc2Vec sentence-vector representation results in those connections to be semantic, i.e. based on meaning. Based on this information, the document of sentences can be represented as a *graph*, like the one in Figure 4.5, where each *node* corresponds to a sentence, and the *edges* represent the connections between them. These connections are weighted, the weights being the inter-sentence similarities defined above.

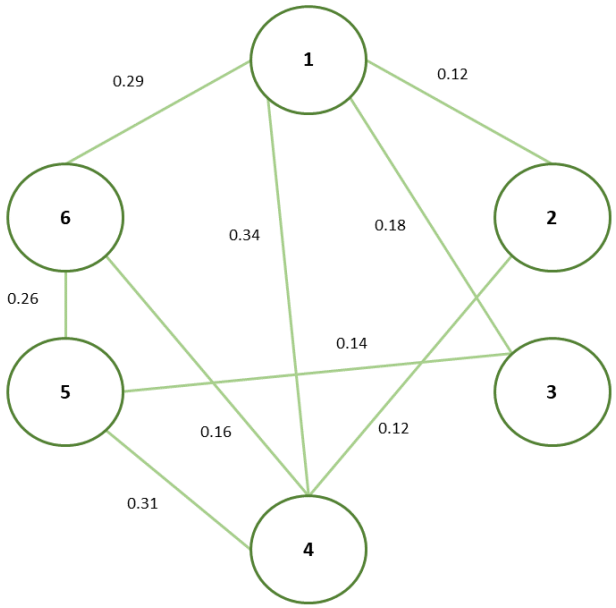


Figure 4.5: Example of a graph with six nodes.

4.3 Ranking Algorithm - TextRank

Once the graph corresponding to the document is constructed, a graph-based ranking algorithm can be applied in order to determine the importance of each node, i.e. sentence. After the sentences are ranked, any number of top K sentences can be extracted to be included in the summary, K depending on the needs of the user.

In this study, the *TextRank* sentence ranking algorithm is used to assign importance to each node (Mihalcea & Tarau, 2004). According to this algorithm, the importance of a node (sentence) of interest depends on three factors: its degree, i.e. the number of nodes this node is linked to by edges (1), the strength of those links determined by the edge weights (2), as well as importance of the nodes the node of interest is linked to. The intuition behind this is that sentences that are highly similar to many other sentences in the document are likely to capture more important information and therefore summarize the main idea of the document well. TextRank computes the importance of the nodes by applying a recursive procedure, since importance scores of the nodes in a graph are dependent on one another.

This approach is based on *PageRank*, Google's algorithm for ranking web pages, where the web pages correspond to graph nodes, and the edges represent the links from one site to another (Brin & Page, 1998). The algorithm models a behavior of a user surfing the web in a random manner, following the links from one web page to another in a long sequence. To avoid the situation where the user is stuck because there are no outcoming links on the page, the user can jump to any random page with probability d, called the damping factor, which is usually set in the range $[0.1, 0.2]$, and can click on a link with probability $(1 - d)$ respectively. In the long run, the proportion of total visits corresponding to a certain page would determine the prestige of the page, i.e. its importance. This probability is called PageRank score, based on which the pages are then ranked. The assignment of importance to nodes can be seen as a probability distribution over the corresponding states, i.e. web pages. Formally, the assignment of importance scores can be seen as a probability vector $\mathbf{p} = (p(1), p(2), \ldots, p(S))$, where $\sum_{n=1}^{S} p(n) = 1$. As the user makes a step (either clicking on a link or jumping to a random page), the probability vector is updated. These updates are performed until convergence is reached, that is, until the fluctuations in probability values become negligible (Manning et al., 2008).

The authors of TextRank have identified that the same idea can be applied to ranking pieces of text. TextRank is different in that the link strength can be incorporated

into the model. That is, the links are weighted, since the links between pieces of text (sentences) can be partial depending on their similarity, while the links between web pages are discrete (either present or not). Moreover, the graphs corresponding to a set of web pages are directed, meaning that the number of incoming links of a node is not necessarily equal to the number of outcoming links, while the links between the sentences are symmetric, making the graphs undirected.

The underlying iterative procedure of TextRank algorithm starts with each node having the same TextRank score (i.e. importance score) - a uniform probability distribution. Thus, the initial probability vector of dimension S would be:

$$\boldsymbol{p_0} = (\frac{1}{S}, \frac{1}{S}, ..., \frac{1}{S}). \tag{4.5}$$

After that, iterations are performed to update each element of the probability vector using the following formula:

$$p(n) = d\frac{1}{S} + (1-d) \sum_{m \in adj[n]} \frac{w_{nm}}{\sum_{k \in adj[m]} w_{km}} p(m) \tag{4.6}$$

where d is the damping factor defined earlier, $adj[n]$ is the set of sentences that are linked to the sentence of interest, $adj[m]$ in turn is the set of sentences linked to those sentences, $p(n)$ and $p(m)$ are the importance scores of the sentence of interest and of every sentence adjacent to it respectively, and w_{ij} is the similarity between sentences i and j. These updates are conducted on each element of the probability vector and stop when convergence is achieved, i.e. when the probability fluctuations become negligible. The final probability vector with each value corresponding to the sentences in the document are the TextRank scores, and the higher the score of a sentence, the more important it is.

The next steps in the summarization process involve sorting the sentences in the document by TextRank scores in descending order, selecting the top K sentences to be included in the summary (can vary depending on the needs of the user), and arranging the extracted sentences in the order they appear in the original text, i.e. sorting by sentence ID.

4.4 Evaluation

Sections 4.1, 4.2 and 4.3 have outlined the steps that were applied on the 20 earnings calls transcripts in order to obtain corresponding summaries. Two algorithms have been applied that differ in the first step of the summarization procedure, i.e. obtaining a sentence-vector representation for each transcript. In the first algorithm, further referred as the *lexical summarization algorithm*, the bag-of-words sentence representation was used, while the second, *semantic summarization algorithm*, made use of Doc2Vec sentence embeddings. The rest of the steps, that is, computing pairwise similarities, constructing a graph, applying a sentence-ranking algorithm and compiling a summary were the same in both algorithms. This resulted in two summaries of equal length for each of the 20 earnings call transcripts, 40 summaries in total. In order to assess the quality of these summaries as well as identify which algorithm produced summaries that are more suitable for the users' needs, an evaluation procedure needs to be in place. Due to the absence of readily available reference summaries against which the summaries produced by the algorithms can be compared, human evaluation has been conducted. Since the summaries of earnings calls transcripts are intended for the use of financial market participants for making investment decisions, assessment of the summaries by the users themselves would best fit the goal. Therefore, a survey was conducted in which each of the 20 questions (each corresponding to one earnings call transcript) involved showing six judges two summaries, produced by each of the two algorithms. For each transcript, the judges were asked to vote which of the two summaries best meets the criterion of 'containing information that is more useful and relevant for making an investment decision about the company that has conducted the earnings call'. This resulted in 120 votes. The order in which the two summaries are displayed in a question was randomized, so that the participants are not able to identify which algorithm produced the summary. All of the six judges are BSc Economics/Finance graduates that work in the area and are therefore familiar with the language of the earnings calls.

In order to assess the outcome of the survey, the proportions of votes cast for each algorithm have been calculated. Significance testing in a form of a *z-test for a single proportion* was then performed to check the claim that the semantic summarization algorithm produces more useful summaries, that is, whether the proportion of votes cast for it is significantly greater than 50%. Therefore, the null and the alternative hypotheses are as follows:

$$H_0 : p = 0.5$$

$$H_a : p > 0.5,$$

where p is the unknown population proportion of votes cast for the summaries produced by the semantic summarization algorithm. The sample proportion is $\hat{p} = \frac{X}{n}$, where n is the sample size (the total number of votes) and X is the number of votes cast for the semantic summarization algorithm. The corresponding z-statistic is as follows:

$$z = \frac{\hat{p} - p_0}{\sqrt{\frac{p_0(1-p_0)}{n}}}, \tag{4.7}$$

Where $p_0 = 0.5$. The corresponding $P - value$ was then obtained and compared against the significance level of $\alpha = 0.05$. If $P - value < 0.05$, it means that the evidence against the null hypothesis is strong enough to reject it, supporting the claim that the semantic summarization algorithm performs better.

Chapter 5

Results

In the previous section, the process of obtaining an automatic summary using a graph-based approach has been outlined and explained. Moreover, two algorithms which employ different sentence-vector representations as part of the summarization process have been defined. The current section will now describe the results that have been obtained by applying the two algorithms on the earnings calls transcripts dataset.

As described in the Methods section, the sentence-vector representations were first obtained from the pre-processed documents. For the lexical summarization algorithm, obtaining the TF-IDF vectors was straightforward as described in Section 4.1.1, while obtaining sentence embeddings for the semantic summarization algorithm required making a number of choices, namely the window size and the dimension of the embeddings. For that, three options for window size (2,3 and 4) and three options for the embedding dimension (25, 50 and 100) were considered, resulting in a total of nine configurations. The optimal Distributed Memory model has been obtained using the procedure outlined in Section 4.1.2. For every configuration, Table 5.1 displays the resulting percentage of times the model has produced embeddings that ranked the sentence as the most similar to itself. It can be seen that all combinations have resulted in the model ranking a sentence as most similar to itself between 94.96% and 96.15% of the times, and mistakenly ranking it as most similar to another sentence 3.85-5.04% of the times respectively. Since the fluctuations in performance are not substantial, it can be said that the model is robust to changes in the two hyperparameters. Combination 7 gave the most consistent result,

with the window size of 2 and embedding vector size of 100 and was thus selected to produce the final Distributed Memory model in order to obtain sentence-vector representations for the semantic summarization algorithm. However, the results in Table 5.1 indicate that any of the given configurations could be selected without a sharp drop in performance. The training of the Doc2Vec model has been conducted on the entire corpus of 3568 sentences corresponding to the 20 transcripts.

Table 5.1: Different hyperparameters configurations of the Doc2Vec model

Combination	**Embeddings dimension**	**Window Size**	**% of times a sentence is ranked as similar to itself**
1	25	2	94.96%
2	25	3	95.04%
3	25	4	95.44%
4	50	2	96.08%
5	50	3	95.88%
6	50	4	96.08%
7	100	2	96.15%
8	100	3	96.01%
9	100	4	95.96%

The next step involved obtaining pairwise similarities between all sentences for each earnings call transcript. Using different sentence-vector representations in the two algorithms implies that the interpretation of the similarities is also different. In the lexical summarization algorithm, where the bag-of-words sentence representation was used, the inter-sentence similarities were expected to be determined based on the common word token they contain (the words that look exactly the same), while in semantic summarization where the sentence embeddings were used, the similarities were expected to be based on meaning, and not necessarily containing the exact same words. To illustrate this, Table 5.2 displays a random sentence drawn from one of the earnings calls transcripts (document 9), as well as a sentence that is the most similar to it based on cosine similarity. The table displays the most similar sentence extracted using bag-of-words representation as well as the most similar sentence obtained using sentence embeddings representation. The sentence of interest presents information about the benefits that the availability of data results in, namely “a greater value proposition for advertisers” and “more informed and efficient content development”. It can be seen that the lexical summarization algorithm returns a sentence that is

similar to the sentence of interest based on the common tokens, that is 'content' and 'advertising'. As for the semantic summarization algorithm, it is interesting that even though the only common token it has with the sentence of interest is 'advertising', the algorithm was still able to extract a sentence that is comparable to the target sentence in terms of meaning. To be precise, the target sentence talking about 'available data' is comparable to the other sentence talking about 'customer insights', and 'greater value proposition to offer advertisers' can be matched with 'advertising opportunities'. Therefore, despite the two sentences having only one common token, the algorithm that uses sentence embeddings was able to identify that the two sentences are relatively close in terms of their meaning.

Table 5.2: An example of a randomly drawn sentence from document 9 and its most similar sentences, according to the lexical and semantic summarization algorithms respectively.

Sentence of interest	**Most similar sentence based on cosine similarity**	
	Lexical summarization algorithm	**Semantic summarization algorithm**
And the vast amount of data available from the combined company will allow not only a greater value proposition to offer advertisers, but will allow more informed and efficient content development.	Owning content will help us innovate on new advertising options, which combined with subscriptions will allow us to grow two-sided business models, help pay for the cost of content creation.	You put that with our customer insights and the addressable advertising opportunities that flow from that, we think we build something here that's really special and it creates significant strategic as well as financial benefits.

The next step in generating summaries involved obtaining the graphs for each document. Figure 5.1 zooms into a random sequence of 20 sentences (sentence IDs 50 to 70) from the document 9, so that the graphs are more readable. The grey edges represent links between sentences with positive weights, while red edges indicate negative weights. The width of an edge is proportional to the absolute value of the corresponding cosine similarity, and an edge is absent if this similarity is zero. On the left-hand side of Figure 5.1, it can be seen that the graph produced as part of

the lexical summarization algorithm is sparser since a lot of the connections between sentences are zero. This is because if the sentences do not have any overlapping words (that is, the term dimensions in which the TF-IDF values are greater than zero in one vector do not overlap with those of the other vector), the cosine similarity between them will be exactly zero. On the other hand, for for the semantic summarization algorithm, the nodes in the graph are highly connected. This is because the sentence embedding vectors are very dense with almost no zeros, making it hard to obtain the cosine similarities that are exactly zero. Another reason is that sentence embeddings go beyond the token-based similarity and also incorporate meaning. Therefore, the sentences will have a connection at least to some extent even if they are completely different in terms of common words, but contain words or phrases that are similar to some extent. This gives an opportunity for sentences that are connected to other sentences through meaning to be assigned a higher importance score. For example, the graphs illustrate that while lexical algorithm completely ignores sentence 55, there is a higher chance for it to be included in the summary according to semantic algorithm. It should also be noted that using bag-of-words representation results in all weights being positive (since all TF-IDF values are positive, cosine similarity ranges between 0 and 1) while using Doc2Vec enables capturing negative relationships between sentences (cosine values range between -1 and 1).This means that when applying the TextRank algorithm, two sentences with opposite meaning can actually lower each other's importance scores.

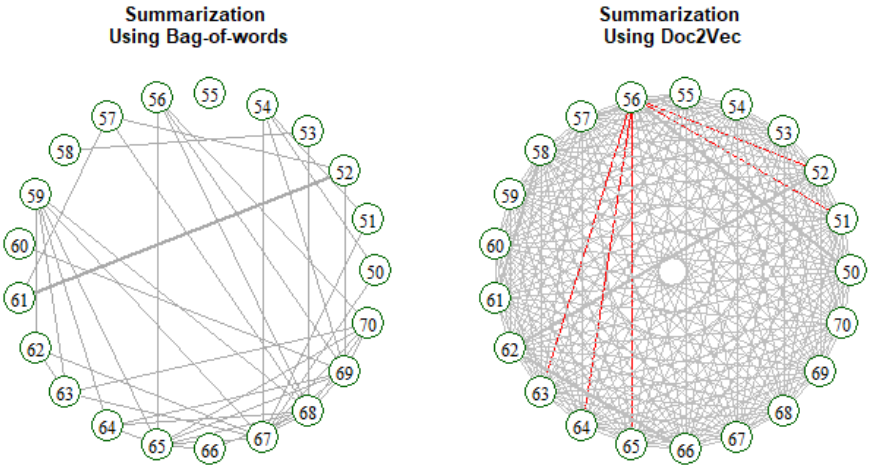


Figure 5.1: Graph corresponding to sentences 50-70 of document 9.

After the graphs were constructed, the TextRank ranking algorithm was applied on each graph in order to obtain the importance scores of the sentences. The damping factor was set to 0.15, like in the original TextRank paper. Once TextRank scores were assigned to each sentence, the top five sentences were extracted for the summary, and the resulting sentences were ranked in the original order (by sentence ID). The resulting summaries were used for evaluation. The choice of including five sentences in the summary is rather arbitrary, but can be justified in that the summary would contain enough sentences to grasp the main idea of the document, but not too many sentences so that the survey respondents are not overwhelmed by large amounts of information. However, the number of sentences in the document can be changed depending on the needs of the reader.

The five-sentence summaries produced for each of the 20 transcripts by the lexical and semantic summarization algorithms are presented side by side in Appendix A (Table A.1). At first glance, both algorithms seem to have produced reasonable summaries containing various information about a firm's financial and operational results (e.g. document 1), mergers and acquisitions (e.g. document 9), expectations for the future (e.g. the semantic summary for document 7) etc. Taking a closer look, however, reveals that the lexical summarization algorithm tends to occasionally include irrelevant transition sentences into a summary, which do not convey any useful information. For example, looking at the summaries for document 8, the lexical summarization algorithm has found sentences like 'Let me turn to slide 10 for a preview of 2017.' and 'Let me turn to slide 13.' important. Similarly, for document 20, the lexical summary contains redundant and uninformative sentences such as 'Going now to slide 5.' or 'So the oil is not gone.'. Moreover, it can be noted that in both examples, the lexical summaries repeatedly include sentences that share certain word tokens, such as 'slide' (document 8) and 'oil' (document 20). This means that occasionally, several sentences in a single lexical summary cover the same subject, which might pose a problem in terms of versatility and comprehensiveness of information that the summary provides.

To examine whether the aforementioned issues as well as other overlooked factors assert themselves in practice, the summaries produced by both algorithms were shown to six individuals in a form of a survey described in Section 4.4. Respondents were asked to vote which algorithm produced a summary that contains information that is more useful for making an investment decision. Six respondents voting 20 times resulted in 120 votes in total. The results have shown that out of the 120 votes, 71 were cast for the summaries generated by the semantic summarization algorithm,

which is equivalent to 59,2% of the votes (Figure 5.2). That is, the respondents found the summaries generated with the use of sentence embeddings slightly more useful. However, one should be careful in concluding that the semantic algorithm performs better for the underlying purpose. Therefore, a z-test for proportions has been preformed as outlined in Section 4.4. Given the outcome ($z = 2.01$, P-value $= 0.02$), it is safe to claim that the semantic summarization algorithm performs better, given the significance level of 0.05, since the resulting P-value is smaller than 0.05.

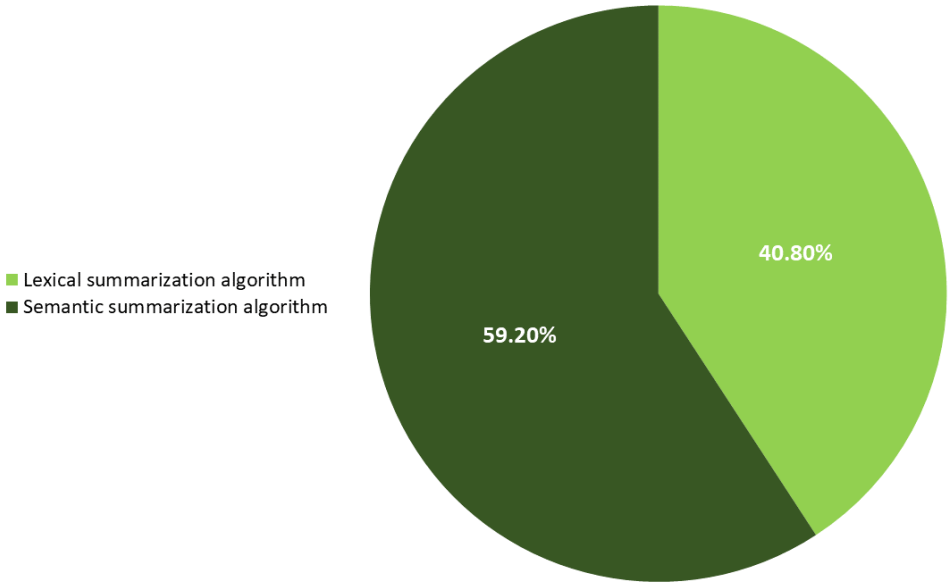


Figure 5.2: Evaluation results: proportion of votes given to the lexical and semantic summarization algorithms respectively.

Chapter 6

Conclusion

This research was conducted with the aim of illustrating the process of graph-based text summarization applied on earnings call transcripts, as well as potentially discovering the benefits of incorporating semantics into this process. The goal of this research was:

To examine whether a graph-based summarization algorithm that incorporates semantic inter-sentence similarities, as opposed to lexical similarities, produces more useful summaries of earnings call transcripts.

To address the research goal, two graph-based summarization algorithms which differ in their sentence-vector representations were compared in this study. One of them, lexical summarization algorithm, uses the bag-of-word model with TF-IDF weighting to build the sentence vectors, and therefore the sentences in the summaries are based on lexical centrality. The second, semantic summarization algorithm, on the other hand, employs the Doc2Vec model to obtain sentence vectors, which results in summaries which contain sentences that are central in terms of their meaning. The two algorithms have been compared by having them applied on 20 earnings call transcripts. First, a qualitative comparison of the summaries produced using the two approaches has revealed that the ones produced by the semantic summarization algorithm contain potentially more useful sentences. This was evident from the semantic algorithm including relatively fewer redundant transition sentences, and more information related to a firm's performance. Moreover, the semantic summaries proved to be less repetitive and therefore provide more diverse information,

which is beneficial in the case of corporate disclosure, since the users would prefer to learn about different aspects of a firm's performance in order to make an investment decision. Evaluating these two algorithms quantitatively with the help of six human judges and performing a z-test for a single proportion has shown that the semantic summarization algorithm has indeed proved to be significantly more useful.

The semantic summarization algorithm can serve as a base for an automatic summarization tool which would take a raw earnings call transcript as an input, and output a summary, which would then be used by a market participant for making an investment decision. However, the semantic algorithm considered in this study has a number of limitations due to the time and resource constraints, making the algorithm a starting point for a more advanced tool that would be more closely tailored to the domain. The following paragraphs will therefore discuss the shortcomings of this study and the further research opportunities that arise from that.

One limitation stems from the fact that the pre-processing steps conducted in this research could be tailored more closely to the domain. This could be one of the causes for one of the algorithms to include transition sentences, which do not convey information that is relevant in the corporate disclosure domain (such as 'Let me turn to slide 13'). It is not entirely clear why the lexical summarization algorithm tends to include such sentences more often, nor it is certain that it is a problem inherent to the lexical algorithm alone. To limit the chance of such sentences being included in a summary by either algorithm, one solution can be to tailor the pre-processing in such a way that these sentences are excluded at the pre-processing stage (e.g. remove the sentences that contain the word 'slide'). Another solution could be to customize the list of stop words to the domain. However, additional research would be needed in order to determine which words are highly irrelevant and could specifically affect the quality of a summary.

Moving on to the second limitation, this research has a narrow scope in that it only investigates one aspect of earnings calls summarization: sentence-vector representation. There are other steps in the summarization process, such as the similarity measure or the sentence ranking algorithm, the tweaking and altering of which can further be studied with respect to earnings calls transcripts. Therefore, further examination in this domain is required in order to be able to build a properly functioning summarization tool for earnings calls transcripts.

Third, in this study, relative usefulness of the two kinds of summaries has been examined and evaluated with the help of potential users. However, it was not in-

vestigated whether the summaries are useful in absolute terms or whether they help mitigate information overload. A suggestion for further research therefore can be to conduct an empirical study in which the effect of availability of a summary on some information overload indicators, such as the volumes of securities trading, would be examined.

The final limitation is related to the absence of high-quality reference summaries. Even though human evaluation of the summaries is desired in this particular domain, since an ideal summary would differ from user to user and their needs, it has still posed some problems in terms of being time-consuming and unautomated. For example, reference summaries would allow to consider, compare and more easily tune more than two algorithms, and arrive at an optimal model in a more efficient manner. This means that a look into what constitutes a good reference summary in the domain of corporate disclosure is needed, which would then potentially lead to a construction of a high-quality labelled dataset, setting the stage for more research in this domain.

Bibliography

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Kochut, K., Trippe, E., & Gutierrez, J. (2017). Text summarization techniques: A brief survey. *International Journal of Advanced Computer Science and Applications*, *8*(10).

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, 1137–1155.

Bowen, R. M., Davis, A. K., & Matsumoto, D. A. (2002). Do conference calls affect analysts' forecasts? *The Accounting Review*, *77*(2), 285–316.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, *30*, 107–117.

Celikyilmaz, A., & Hakkani-Tur, D. (2010). A hybrid hierarchical model for multi-document summarization., In *Proceedings of the 48th annual meeting of the association for computational linguistics*.

Dumais, S. T., Furnas, G. W., Landauer, T. K., Deerwester, S., & Harshman, R. (1988). Using latent semantic analysis to improve access to textual information, In *Proceedings of the sigchi conference on human factors in computing systems*.

Dyer, T., Lang, M., & Stice-Lawrence, L. (2016). Do managers really guide through the fog? on the challenges in assessing the causes of voluntary disclosure. *Journal of Accounting and Economics*, *62*(2-3), 270–276.

Edmundson, H. (1969). New methods in automatic extracting. *Journal of the ACM*.

Erkan, G., & Radev, D. R. (2004). Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, *22*, 457–479.

Fama, E. (1970). Efficient capital markets: A review of theory and empirical work. *The Journal of Finance*, *25*(2), 383–417.

Frankel, R. M., Johnson, M. F., & Skinner, D. J. (1999). An empirical examination of conference calls as a voluntary disclosure medium. *Journal of Accounting Research*.

Gambhir, M., & Gupta, V. (2017). Recent automatic text summarization techniques: A survey. *Artificial Intelligence Review*, *47*(1), 1–66.

Gupta, V., & Lehal, G. S. (2009). A survey of text mining techniques and applications. *Journal of Emerging Technologies in Web Intelligence*, *1*(1).

Healy, P. M., & Palepu, K. (2001). Information asymmetry, corporate disclosure and the capital markets: A review of the empirical disclosure literature. *Journal of accounting and economics*.

Hollander, S., Pronk, M., & Roelofsen, E. (2010). Does silence speak? an empirical analysis of disclosure choices during conference calls. *Journal of Accounting Research*, *48*(3), 531–563.

Kågebäck, M., Mogren, O., Tahmasebi, N., & Dubhashi, D. (2014). Extractive summarization using continuous vector space models., In *The 2nd workshop on continuous vector space models and their compositionality (cvsc)*.

Lee, C., Jian, Z., & Huang, L. (2005). A fuzzy ontology and its application to news summarization. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, *35*(5), 859–880.

Luhn, H. (1958). The automatic creation of literature abstracts. *IBM Journal of Research and Development*, *2*(2), 159–165.

Manning, C., Raghavan, P., & Schütze, H. (2008). Link analysis, In *Introduction to information retrieval*, Cambridge University Press.

Mihalcea, R., & Tarau, P. (2004). Textrank: Bringing order into texts., In *Proceedings of the 2004 conference on empirical methods in natural language processing*.

Mikolov, T., & Le, Q. (2014). Distributed representations of sentences and documents., In *International conference on machine learning*.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Miller, B. (2010). The effects of reporting complexity on small and large investor trading. *The Accounting Review*, 2107–2143.

Mitchell, T. (1997). *Machine learning*. Burr Ridge, IL, McGraw Hill.

Nenkova, A., Vanderwende, L., & McKeown, K. (2006). A compositional context sensitive multi-document summarizer, In *Proceedings of the 29th annual international acm sigir conference on research and development in information retrieval - sigir '06*.

Paredes, T. A. (2003). Blinded by the light: Information overload and its consequences for securities regulation. *Washington University Law Quarterly*, *81*(2), 417–486.

Plaza, L., Díaz, A., & Gervás, P. (2011). A semantic graph-based approach to biomedical summarisation. *Artificial Intelligence in Medicine*, *53*(1), 1–14.

Řehůřek, R. (2019). *Gensim: Doc2vec model*. https://radimrehurek.com/gensim/auto_examples/tutorials/run_doc2vec_lee.html

Salton, G., Singhal, A., Mitra, M., & Buckley, C. (1997). Automatic text structuring and summarization. *Information Processing Management*, *33*(2), 193–207.

Steinberger, J., & Jezek, K. (2004). Using latent semantic analysis in text summarization. *Proc. ISIM 4*, 93–100.

Wang, K., Wu, M., & Li, W. (2008). Extractive summarization using supervised and semi-supervised learning, In *Proceedings of the 22nd international conference on computational linguistics (coling 2008)*.

Appendix A

The Summaries

Table A.1: Summaries produced using two different sentence-vector representations

ID	Lexical summarization algorithm	Semantic summarization algorithm
1	Gross margin is up 7.3% up to €9.5 billion, driven by 12.1% increase in Networks, and 19.3% growth in Renewables. In the UK, EBITDA reached £182 million. In Mexico, EBITDA grew 11.5% to $379 million due to the renegotiation of old contracts with a negative impact last year of $66 million. Renewables EBITDA increased by 22.7% to €1,126 million, driven by the recovery in Spain and the positive performance in the UK. The €939 million FX impact, negative impact on our debt is more than compensated by the €1 billion positive impact due to the stronger cash flow generation and tariff deficit securitization.	EBITDA grew 5.8% and recurring net profit 8.5%. Gross margin increased 7.3% to €9.5 billion as revenues grew more than procurements. Net operating expenses rose 13.9% including non-recurring items and accounting reclassifications in the U.S. In the UK, EBITDA reached £182 million. EBITDA grew 1.1% to €3,027 million.

Continued on next page

Table A.1 – continued from previous page

ID	Lexical summarization algorithm	Semantic summarization algorithm
2	From a brand point of view, growth of our Top 14, 2%; growth of our Priority Premium Wines, 6%; growth of our Key Local Brands, 6%. As Alexandre said, organic growth of top line 3%, organic growth of profit from recurring operations up 3%. So you have 3% organic growth I just mentioned. For the full year 2015/2016, we expect a positive forex impact, based on the current rates of €20 million on the profit from recurring operations. It was 4.6% last year.	Decline in France and Russia mainly due to technical impacts. Strong performance of Jameson, of Martell, Glenlivet, PJ, Mumm and our Indian whiskies. I'll mention also Absolut and some other brands like Ballantine's. Very good performance on Absolut, Glenlivet, Jameson and some of our wine brands, principally Jacob's Creek. Again, Noblige La French Touch limited edition.
3	During the quarter, we completed a $1.2 billion unsecured credit facility. Our new same-store pool will increase by a 168 stores for a new total of 732. We also project $225 million in joint venture acquisitions with approximately $75 million in capital to be contributed by Extra Space. I will now turn the time back to Joe. We expect 2017 same-store revenue growth and NOI growth in the 4% to 5% range, which we believe will be better than nearly all the other state sectors.	FFO per share as adjusted increased by 18%. To date, we have drawn $662 million. We are also managing many newly-constructed assets on a third-party basis, which provides fee income, strengthen our brand, and increase our scale. Second, demand is steady. Third, we have other tools that contribute to our FFO growth.

Continued on next page

Table A.1 – continued from previous page

ID	Lexical summarization algorithm	Semantic summarization algorithm
4	Turning to the results for the group, BP's first quarter underlying replacement cost profit was $530 million, down 79% on the same period a year ago and 170% higher than the fourth quarter of 2015. In the Downstream, the first quarter underlying replacement cost profit before interest and tax was $1.8 billion compared with $2.2 billion a year ago and $1.2 billion in the fourth quarter of 2015. The Fuels business reported an underlying replacement cost profit before interest and tax of $1.3 billion in the first quarter compared with $1.8 billion in the same quarter last year and $890 million in the fourth quarter of 2015. In Other Business and Corporate, we reported a pre-tax underlying replacement cost charge of $180 million for the first quarter, $110 million lower than the same period a year ago. The group's cash costs over the last four quarters were $4.6 billion lower than 2014.	The global refining marker margin averaged $10.50 per barrel in the first quarter, the lowest since the third quarter of 2010, weighed down by weak diesel demand and high gasoline stocks in the United States. In the Downstream, the first quarter underlying replacement cost profit before interest and tax was $1.8 billion compared with $2.2 billion a year ago and $1.2 billion in the fourth quarter of 2015. The pre-tax cash outflow on costs related to the oil spill for the first quarter was $1.1 billion, including $530 million relating the 2012 criminal settlement with the United States Department of Justice. We also wish to retain flexibility to add to the portfolio at the lowest point of the cycle if the right opportunities present themselves. In Oman, we signed a major agreement to extend the Khazzan license to access a further 1,000 square kilometers, estimated to contain around 3.5 trillion cubic feet of gas.

Continued on next page

Table A.1 – continued from previous page

ID	Lexical summarization algorithm	Semantic summarization algorithm
5	And for the first six months of 2016, the operating result came to €626 million. The operating result for Netherlands Non-life decreased to €19 million from €45 million for the second quarter of 2015. The operating result of Japan Life was €23 million in the second quarter of 2016, down from €25 million in the second quarter of 2015. The total operating result of the segment Other improved to €2 million in the second quarter of 2016 from a loss of €7 million in the second quarter of 2015. Finally, the operating result of NN Bank increased to €17 million in the second quarter of this year from €6 million in the same quarter of 2015.	And NN Bank again reports healthy growth in both mortgages and savings. The Solvency II ratio of NN Group increased to 252% which already reflects deductions for the €500 million share buyback program and the 2016 interim dividend. Expenses also decreased, reflecting lower staff-related expenses, as well as lower volume-driven fixed service fees. The cost-income ratio increased slightly as fees decreased more than expenses. Let's look at this individually.

Continued on next page

Table A.1 – continued from previous page

ID	Lexical summarization algorithm	Semantic summarization algorithm
6	So, Americas, last year were up 2%, this year, it's 4% driven by the U.S. Last year was up 2%. Innovation has driven growth this year. Last year, they were down 1%. For this new fiscal year, fiscal year 2017, our objective is to improve, obviously, our performance in China.	We'll then give you a chance to ask your questions. We started some years ago, I think it was three years ago, to invest behind our premium brands, our growth relays with the likes of Ballantine's Finest, Absolut and our champagne brands. And there has been a slightly slower trend in the second half of our fiscal year with a slight deceleration in the mix. Very good performance of Beefeater across Europe, Spain obviously, but across the rest of Europe as well. This leads us to give a guidance of organic growth in profit from recurring operations comprised between plus 2% and plus 4%.

Continued on next page

Table A.1 – continued from previous page

ID	Lexical summarization algorithm	Semantic summarization algorithm
7	On slide 13, you'll see revenue for the motorcycles and related products segment was down in the third quarter behind a weak U.S. retail industry and our corresponding decrease in year-over-year motorcycle shipments. On slide 15, operating margin as a percent of revenue for the quarter was 10.0%, down compared to last year's third quarter. During the quarter, HDFS' operating profit decreased $3.4 million or 4.6% compared to last year. We are very encouraged by the momentum we experienced in the U.S. in September when our new Model Year 2017 motorcycles drove an increase in retail sales of approximately 5% and a more than 3 percentage point increase in market share. Overall, we are thrilled with the initial response to our new Model Year 2017 motorcycles despite the weakness in the U.S. industry retail sales.	As we indicated on our last call, we will continue to manage supply in line with demand. We're also investing to reach new riders by growing distribution internationally and by training new riders right here in the United States. On slide 11, you'll see retail sales in our international markets were up 1.0% in Q3. Operating margin was impacted by lower gross margin as well as higher SG&A as we increased our investments in demand-driving and product development. We expect worldwide retail sales growth in the fourth quarter driven by significant demand for our Milwaukee-Eight engine, our increase in demand driving investments, expansion of the international dealer network, and lapping last year's U.S. industry decline of 3.8%.

Continued on next page

Table A.1 – continued from previous page

ID	Lexical summarization algorithm	Semantic summarization algorithm
8	Finally, in Safety and Productivity Solutions, sales were down 8% on a core organic basis. I have a later slide on what these OEM incentives mean for our future growth. Let me turn to slide 10 for a preview of 2017. Let me turn to slide 13. We'll see the benefits of this next year, 2018, and many years to come.	And as always, we'll leave time for your questions at the end. As we previewed during our update call two weeks ago, for the third quarter we reported earnings per share of $1.60 or $1.67 excluding the $0.07 we deployed to restructuring. This is also supported by a slightly easier setup in industrial safety, driven by prior-year oil and gas related declines. With more than 1.9 billion share repurchases this year, our share count will be approximately 1% lower in the fourth quarter. We've talked extensively over the past few years about our capital expenditures.
9	The deal also improves our strong free cash flow dividend coverage. All this continues to drive strong cash flows. We had our second highest ever operating cash flows of $11 billion, with free cash flow reaching $5.2 billion. We added 1.5 million U.S. subscribers at positive phone net adds and grew our smartphone base by 700,000. And free cash flow growth has been strong with dividend coverage year-to-date at 67%.	As you know, on Saturday we announced an agreement for AT&T to acquire Time Warner. Time Warner is the global leader in media and entertainment with terrific brands and their brands we all know and love, from Game of Thrones to CNN and Superman, just to name a few. And together with AT&T, we'll develop new innovative business models and forms of content that consumers will be demanding tomorrow in this ubiquitous, multi-platform, on-demand, and increasingly mobile environment. So now I'll turn it over to John Stephens. That's up 9.1% and when you adjust for APEX, growth was closer to 10%.

Continued on next page

Table A.1 – continued from previous page

ID	Lexical summarization algorithm	Semantic summarization algorithm
10	So at this time, I would like to turn the time over to Joe. It was another strong quarter for Extra Space. The largest of the off-market transactions closed on September 16 when we purchased Prudential's majority interest in 23 stores for $238 million. We also expect to close $255 million in CofO acquisitions in 2016. $90 million of these will be wholly-owned and the remainder will be in joint ventures with our investment in these ventures totaling $53 million. Subsequent to the end of the quarter, we completed a $1.15 billion unsecured credit facility.	The company assumes no obligation to revise or update any forward-looking statements because of changing market conditions or other circumstances after the date of this conference call. We are focused on using all of these levers to continue to grow shareholder value. Discounts, while still very low from our historical measure, are up from an all-time low in 2015. Quarter end occupancy was strong at 93%. Last night, we reported FFO as adjusted of $1.02 per share exceeding the high end of our guidance by $0.02.

Continued on next page

Table A.1 – continued from previous page

ID	Lexical summarization algorithm	Semantic summarization algorithm
11	The third quarter underlying replacement cost profit before interest and tax was $1.4 billion compared with $2.3 billion a year ago and $1.5 billion in the second quarter. The Fuels business reported an underlying replacement cost profit before interest and tax of $1 billion compared with $1.9 billion in the same quarter last year and $1 billion in the second quarter of 2016. The Lubricants business reported an underlying replacement cost profit of $370 million in the third quarter compared with $410 million in the second quarter and $350 million a year ago. Relative to 2016, we expect 2017 operating cash flow to benefit from improved environment that would add an incremental $2 billion to $4 billion to cash flow. As we move through 2017, we also expect operating cash flow to be supported by growth and continued underlying performance improvements in both our businesses.	Oil prices fell in July as oil inventories reached a new high, but improved again towards the back of the quarter on improving fundamentals further supported by the newly stated intentions of OPEC. Third quarter underlying operating cash flow, which excludes Gulf of Mexico oil spill payments, was $4.8 billion. Turning to the Gulf of Mexico oil spill costs and provisions. We now expect capital expenditure to be around $16 billion this year compared to our original guidance of $17 billion to $19 billion. In Petrochemicals, we launched the new low-carbon brand of PTA, which through our proprietary technology, supports around a 30% lower carbon footprint than the average European PTA production.

Continued on next page

Table A.1 – continued from previous page

ID	Lexical summarization algorithm	Semantic summarization algorithm
12	Turning to our core operating performance for the year. We continue to see generally solid overall demand for our products and services across our commercial, defense and space and services businesses. Revenue for the year was a strong $94.6 billion on solid commercial airplane deliveries and continued growth in our services business. For the fourth quarter, our commercial airplane revenue was $16.2 billion. Our 2017 commercial airplane revenue guidance is between $62.5 billion and $63.5 billion.	In summary, our team is intent on building upon the strategic progress and business performance momentum established over the past several years to meet our commitments to our stakeholders and accelerate improvements in quality, safety and productivity, all to drive further innovation in our products and processes and deliver long-term growth and value creation for our company. Now over to Greg for our financial results and our 2017 guidance. Operating cash flow for the year was a record $10.5 billion. On the 787 program, the deferred production balance continued its downward trend, declining another $215 billion in the quarter. Defense, space and security reported $8 billion of new business in the quarter and the backlog now stands at $57 billion, of which, 37% represents customers from outside the United States.

Continued on next page

Table A.1 – continued from previous page

ID	Lexical summarization algorithm	Semantic summarization algorithm
13	Gross margin as a percent of revenue decreased versus prior year quarter, as a result of unfavorable currency exchange. For the full year, worldwide retail sales were down 1.6% compared to last year. During the quarter, HDFS' operating profit decreased $0.7 million or 1.2% compared to last year. For the full year, HDFS continued to have a strong U.S. retail market share of new Harley-Davidson motorcycle sales at nearly 62%. For the full-year 2017, we expect operating margin as a percent of revenue for the Motorcycles segment to be approximately in line with 2016 operating margin.	For the full year, our market share was up 1.0 percentage points to 51.2%. We are very pleased that we were able to grow our market share in the U.S with our brand-enhancing actions and product innovation, despite high levels of discounting by the competition. Sales were down in the region largely due to continued declines in Brazil resulting from a slowing economy, consumer uncertainty, and very aggressive price competition. We expect our pricing actions to be largely offset by unfavorable currency exchange, higher raw material costs and increased manufacturing expense. Our investments in new product development will enable us to reinvent the product segments we compete in today and push us into new segments.

Continued on next page

Table A.1 – continued from previous page

ID	Lexical summarization algorithm	Semantic summarization algorithm
14	We're committed to building riders. As expected, U.S. retail sales were down compared to last year's first quarter. We expect worldwide retail sales growth for the remainder of the year will be driven by improved availability of Milwaukee-Eight powered bikes and the introduction of our model year 2018 motorcycles, strong execution of our U.S. ridership growth initiatives, expansion of our international dealer network and easing of year-over-year sales comps in both the U.S. and international markets. U.S. retail sales were adversely impacted by weak industry sales and limited availability of model year 2017 motorcycles, partially offset by strong sales of our Milwaukee-Eight Touring bikes and our focus on growing ridership. During the quarter, we expected lower year-over-year U.S. retail sales and stable market share.	We're here to build the next generation of Harley-Davidson riders, ignite their dreams of personal freedom however they define it and we're on it. So with that, have John go through the detail on the first quarter. This action significantly limited our model year 2017 motorcycle availability in the quarter, in particular, our Milwaukee-Eight powered bikes. Worldwide retail sales of new Harley-Davidson motorcycles in Q1 are summarized on slide 13. Operating margin was unfavorably impacted by lower gross margin, partially offset by lower SG&A.

Continued on next page

Table A.1 – continued from previous page

ID	Lexical summarization algorithm	Semantic summarization algorithm
15	Organic sales were up more than 2%. Aerospace segment margin expansion for the quarter of 90 basis points also exceeded the high end of our guidance, driven by productivity, commercial excellence, and the favorable impact of the divestiture of the Government Services business in 2016. Home and Building Technologies generated organic sales growth of 3%, driven by a strong performance in environmental and energy solutions, security and fire, and our global distribution businesses. Organic sales growth is anticipated to be flat to 2%, with 50 to 80 basis points of margin expansion. Our end markets continue to improve across our businesses and our execution is getting better as well.	The mix dynamics of sales in the quarter were a bit less favorable than we anticipated. Performance Materials and Technologies had a very strong quarter. There continues to be increasing interest in domestic modular units in particular. Growth in our IoT business was also strong with good performance in a number of regions, and Intelligrated grew in excess of 20% this quarter compared to the first quarter of 2016 when it was not owned by Honeywell, and this was driven by large projects in a number of key accounts. In summary, we delivered a high quality first quarter result with all of our segments contributing to the performance.

Continued on next page

Table A.1 – continued from previous page

ID	Lexical summarization algorithm	Semantic summarization algorithm
16	In the first quarter of the year, net profit reaches €820 million with an EBITDA of €1,862 million, driven mainly by the contribution of our [ph] novel (1:33) businesses. EBITDA of €1,862 million has mainly been driven by Networks, which increases by 9.1% as result of our investment in that businesses. Getting into the details, in Spain, the EBITDA fell 19.6% to €238 million, mainly driven by a 6.1% lower output with a 41% decrease in hydro production after an exceptional Q1 2016 and lower gas results due to higher procurement costs. In Spain, EBITDA reached €146 million, 16.3% lower than last year as a result of 17.3% lower output compared to an exceptional Q1 2016 that normalized thereafter. Net operating expenses are up £15 million, or 176%, of which £12 million are non-recurring.	This merger has led to the creation of a global leader in the wind generation industry with €11 billion of revenues, €1.1 billion of operating profit, 26,000 employees and with the headquarters in Spain. Also, our American company, AVANGRID, where we hold 81.5% stake, increased net income by 13% in the period, to $239 million. Please remind that in Q1 2016, results have been restated due to the change in the accounting treatment of subsidies from May 16 onwards. In addition, from 2017 onwards, social bonus is accounted for in the supply business instead of the corporation. Reported net profit fell 4.7% to €127 million (sic) [€827 million] (21:43) due to lower operating results and higher non-debt-related financial expenses, partially compensated by the accounting of €255 million net after taxes of the Gamesa merger, of which €198 million are cash from the dividend paid in April and another €58 million are accounted as a sort of asset revaluation.

Continued on next page

Table A.1 – continued from previous page

ID	Lexical summarization algorithm	Semantic summarization algorithm
17	As a result, same-store NOI grew 9.2% and FFO per share as adjusted increased by 20%. Our 2017 same-store pool increased by 168 stores for a total of 732. Due to the Q1 expense beat, we are lowering our annual expense guidance to 2.25% to 3.25%. The mix has changed slightly and now includes $325 million in wholly-owned stores and $190 million in joint venture acquisitions and developments, with approximately $75 million in capital to be contributed by Extra Space. While certain stores and markets have felt the impact of new development, it has not prevented us from experiencing positive revenue growth.	We demonstrated great expense control with a 2% decrease in same-store expenses. Property taxes and payrolls were lower than expected, and utilities and snow removal were below budget due to a mild winter. Same-store NOI benefited 220 basis points from the change in pool during the quarter. This quarter, we are presenting the impact of this change in our Q1 supplemental financial information, which shows the results of our 2016 and 2017 same-store pools with and without tenant reinsurance. As a result, we are increasing our annual NOI guidance to 4.25% to 5.75%.

Continued on next page

Table A.1 – continued from previous page

ID	Lexical summarization algorithm	Semantic summarization algorithm
18	BP's first quarter underlying replacement cost profit was $1.5 billion, around $1 billion higher than the same period a year ago and $1.1 billion higher than the fourth quarter of 2016. The first quarter underlying replacement cost profit before interest and tax was $1.7 billion, compared with $1.8 billion a year ago and $880 million in the fourth quarter. Our estimates of BP's share of Rosneft's production for the first quarter is 1.1 million barrels of oil equivalent per day, an increase of 11% compared with a year ago, and roughly flat compared with the previous quarter. Now looking at cash flow. By the end of this year you should expect to see the momentum in stronger underlying operating cash flows, which, coupled with our focus on capital discipline, will grow capacity to deliver sustainable and attractive returns to shareholders over time.	The first quarter dividend payable in the second quarter of 2017 remains unchanged at $0.10 per ordinary share. The increase versus last year reflects the completion of the acquisition of Bashneft, commencement of the Suzun and East Messoyakha fields, and Rosneft's increased stake in the Petromonagas joint venture. Net debt at the end of the quarter was $38.6 billion dollars, and gearing was at 28%, within our 20% to 30% target band. The first of our seven planned 2017 start-ups, Trinidad Onshore Compression, came online in April. In line with this we recently announced our intention to divest our interest in the SECCO joint venture in China.

Continued on next page

Table A.1 – continued from previous page

ID	Lexical summarization algorithm	Semantic summarization algorithm
19	This increase was driven by the €49 million contribution of the Delta Lloyd businesses, as well as higher results at most segments, partly offset by the impact of a €40 million strengthening of P&C liabilities at Netherlands Non-life. The operating result of Netherlands Life increased to €290 million in the second quarter of 2017 of which Delta Lloyd contributed €57 million. The operating result in disability and accident was €29 million for the second quarter of 2017, of which Delta Lloyd contributed €6 million. The operating result of the segment Other decreased to a loss of €7 million, of which, €5 million related to Delta Lloyd. The operating result of the banking business increased to €34 million, of which, Delta Lloyd contributed €9 million.	In addition, as Delfin will discuss later, the Solvency II ratio includes the alignment of the Delta Lloyd balance sheet assumptions. This was a lot of work but it went very smoothly. This transaction was an important step in reducing leverage and improving our carrying capacity. Turning to slide 19, you can see there that as we already flagged in December last year, we anticipated and have made significant adjustments to the valuation of assets and liabilities on the Delta Lloyd balance sheet, above and beyond those taken by Delta Lloyd in the fourth quarter of 2016 and the first quarter of 2017. Given the larger asset base of our asset manager, we also took the opportunity to reduce the fees paid by NN Life.

Continued on next page

Table A.1 – continued from previous page

ID	Lexical summarization algorithm	Semantic summarization algorithm
20	Going now to slide 5. So the oil is doing very, very well. The oil is not gone here from the 30 POPs. So the oil is not gone. And in particular, what you can see is that the GOR increases we've seen are higher than what we expected.	Production increased 12,000 BOE a day or about 6% quarter over quarter. Incidentally, they're right on track with their oil production. And we can manage through really any downturn or any price scenario, and work towards living more within our cash flow and improve returns by doing so, especially as we drill these high-return wells. And the remaining two POPs remaining in 2017, both will be in the northern part of our acreage. Of those barrels, we are exporting about 1 million barrels per quarter, as Tim talked about, with the rest being sold into the refinery market.