



University
of Glasgow | School of
Computing Science

Inferring Related Companies from Online Publications

Lukas Rubikas

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8QQ

A dissertation presented in part fulfilment of the requirements of the
Degree of Master of Science at The University of Glasgow

December 6th, 2021

Abstract

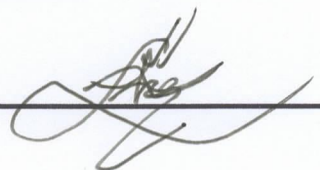
Stock market has become accessible for retail investors more than ever. These new stock market participants need to constantly stay informed in order for their investments to become profitable. The most obvious way to do this is to constantly be aware of on-going stock market trends, keep track of financial stock market movements and read associated stock market news. But what if these types of information don't directly mention any financial companies? In this work I will explore whether it's possible to automatically detect related companies from online publications, be it blog articles, news articles or articles from social media. User study results express a generally positive sentiment to the proposed solution results. Given the work described in this dissertation, providing a solution for automatically finding related companies in online publications is possible, but there are areas where improvements could be made.

Education Use Consent

I hereby give my permission for this project to be shown to other University of Glasgow students and to be distributed in an electronic format. **Please note that you are under no obligation to sign this declaration, but doing so would help future students.**

Name: Lukas RUBIKAS

Signature: _____

A handwritten signature in black ink, appearing to be 'Lukas Rubikas', written over a horizontal line.

Acknowledgements

For the eternal patience, guidance, friendship and endless support I would like to thank the following people:

Una Marie Darragh
Shona Robertson
Richard Mccreadie
Helen Purchase
Moritz Schlichting
Liam Tracy
Jasmine Urquhart
Airi Osaki
Dominykas Staugaitis
Monika Bagaslauskaitė
Daumantas Pagojus
Domantas Jurkus
Akvilė Gečaitė
Rytis Daškevičius
Tomas Liutvinas
Deividas Lenkus
Vladislovas Kofyrinas
Paulius Veliulis (Paul A)
Paulius Kulbokas (Paul One)
Ignas Gaižauskas
Lukas Artūras Gudas
and others.

Contents

1	Introduction	4
2	Related work	5
2.1	Competing products	7
3	Requirements	8
3.1	Non-functional requirements	8
3.2	Functional requirements	8
4	Design & Implementation	10
4.1	Design considerations	11
4.2	Architecture overview	14
4.3	Implementation details	14
5	Evaluation	19
5.1	Cranfield-style ranking & model evaluation	19
5.2	User study & model validation	20
5.3	Requirements Validation	27
6	Conclusion	29
6.1	Future work	29
A	NIO: Consumers Will Soon Realize NIO Has What Other EVs Are Missing, BaaS	30
A.1	Article excerpt	30
A.2	User study evaluation	32

Chapter 1

Introduction

The influence of the retail investors can no longer be deniable. More and more hobbyists invest in stocks or cryptocurrencies every day - it only takes a few minutes to set up an investment portfolio on any personal finance or stock market trading app, like Revolut, RobinHood, or Fidelity.

But as with all personal finance investment types, for a new investor, investing in stock market requires a substantial amount of knowledge, ability and time to consume various information sources (like news articles surrounding particular stocks or economic sectors and industries) and to do so typically fast, in order to have time to react to ever ongoing stock market changes.

But what if the news article that have caught the interest of such new investor doesn't discuss any particular stock at all? What if they only discussed some specific topics, like semiconductor shortage and how it affects the automotive industry, commodities like oil and how the trends around are changing, or some ground-breaking investigative journalism-style story about fast fashion? Wouldn't it be nice to have solution, which would automatically infer a list of stocks that could be invested in or sold, or otherwise analyzed by external solution should substantial news story about the industry in question arise? To have a machine-learning-based solution that doesn't have to be retrained should new stocks (and hence, new labels) arise and become publicly traded?

It is precisely the research question I shall attempt to answer in this dissertation, in which I shall invite the reader together with me to:

- **Review the related work**, like previous natural language processing techniques that were used in similar tasks like this before and the ones that lead to the tools we shall be using.
- **Define functional and non functional requirements**, like what tools should I be using or what exactly the proposed system should do and what capabilities it must posses.
- **State design choices and describe the implementation processes** and various nuances the implementation details contain.
- **Define evaluation procedures and experimental set-up** for evaluating entity inferral performance
- And finally **state conclusions and share ideas for future work** for how to improve the proposed system and various experimental phases and components in the future.

Chapter 2

Related work

Text clustering and classification is considered a classic problem in natural language processing (NLP) tasks. Perhaps the most simplest, nowadays considered a classical approach for fixed-length text representation was introduced as early as 1954 as the bag-of-words (BOW) or the bag-of-n-grams [11] model. In the bag-of-words approach, only the presence and the number of occurrences of words are considered, with no regard for word ordering or document compositionality. Thus, many documents could have exactly the same representation as long as they contain the same terms. Bag-of-words models also suffer from the document collection needing to be defined at training time and thus suffers from high dimensionality and data sparsity. In order to combat these shortcomings, distributed ways of representing text have been proposed [28] as early as 1986 and became an important stepping stone for statistical language modelling, especially for later approaches based on neural networks, now commonly referred as neural (network) language models (NNLM) [1].

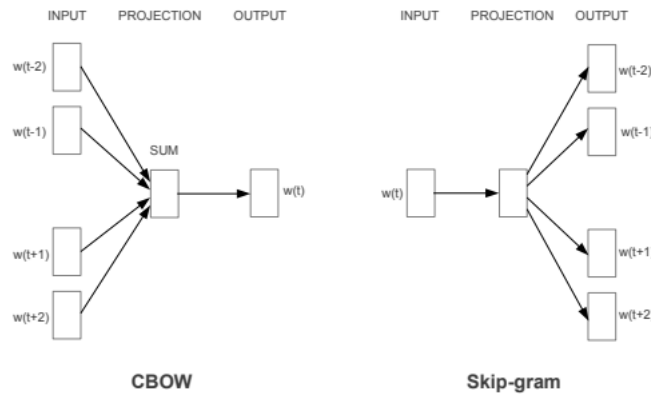


Figure 2.1: Principal Word2vec model architectures. (*Left*) Continuous bag-of-words (CBOW) model architecture, similar to feed-forward neural network language model, where surrounding context window word vectors are either summed, averaged or concatenated to predict the center word. (*Right*) Continuous skip-gram model, where current word vector is used as an input to log-linear classifier with continuous projection layer to predict words surrounding the center word in a small context window.

An important stepping stone for distributed text representation came in 2013, when Mikolov et al. introduced a word-embedding technique called Word2vec [22][23]. The Word2vec algorithm uses

a shallow neural network to be trained on either context word prediction or center word prediction tasks. Vectors inferred from the learned models can be interpreted as geometrical vectors in linear space and compared to each other either by the distance from each other or the angle between them. Word2vec encodes information about word meaning (regarding a small context window in which it appears) and enables the user to perform linear algebra operations, such as the classical “vector(“King”) - vector(“Man”) + vector(“Woman”) = ?” problem ¹.

A natural extension to Word2vec word embeddings came in 2014, when Quoc Le and Tomas Mikolov introduced distributed representations of sentences and documents, known as Doc2vec [17] technique. Doc2Vec model architectures borrow heavily from previous neural network language models, where word vectors belonging in a close context window were averaged or concatenated with each other to predict the next word in a context window. Two principal Doc2Vec model architectures were introduced, paragraph vector - continuous bag-of-words (PV-CBOW) and paragraph vectors - distributed memory (PV-DM).

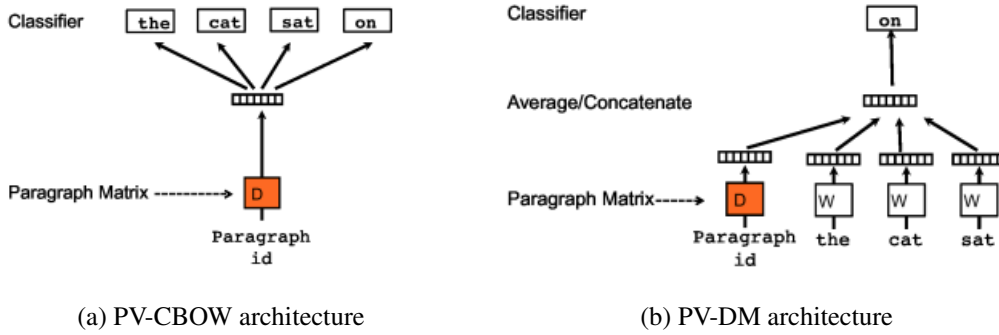


Figure 2.2: Doc2Vec model architectures

In PV-CBOW architecture, word vectors are not trained and only paragraph vectors are trained. Word ordering is disregarded and model simply tries to predict randomly sampled words from the paragraph in the output. This model architecture is conceptually simpler and requires shorter training time and less data to be stored (as word vectors are not trained). It is similar to the Skip-gram model from Word2vec (Figure 2.1).

In PV-DM architecture, paragraph vector trained alongside word vectors and retained throughout all context windows in the document and can interpreted as simply an additional token, representing what is missing from the context window given the whole document. It is either summed, averaged or concatenated with other word vectors in each context window on the next word prediction task. PV-DM architecture is (perhaps confusingly) conceptually similar to CBOW architecture from Word2vec (Figure 2.1).

The original researchers noted that for most tasks, although taking more time to train and requiring more data to store (as word vectors are also trained in addition to paragraph vectors), PV-DM architecture proved to be superior although a combination of both PV-DM and PV-DBOW are more consistent across tasks they have tested and is generally a recommended approach. Both PV-CBOW and PV-DM architecture models can be trained with either hierarchical softmax optimized with Huffman binary tree [24], or noise contrastive estimation (NCE) which was applied to language

¹ Although the intended answer by the original researchers is vector(“Queen”) and of course the outcome depends on the text corpus Word2vec models are trained on, some researchers noted that the closest answer to this problem is almost always the original input vector for “King”. Vector(“Queen”), on the other hand, was almost always the second closest answer.

modelling by Mnih and Teh [25] and when simplified for Do2Vec purposes simply referred as negative sampling technique (NEG).

2.1 Competing products

Not much is known about competing products in a sense of their ability to infer related companies without explicitly mentioning them. Many financial news websites, such as Yahoo Finance, Seeking Alpha, MarketWatch, The Motley Fool and Zacks explicitly list companies that are related to the topic at hand, probably using automatic linking tools in the article writing software or linking topic companies manually.

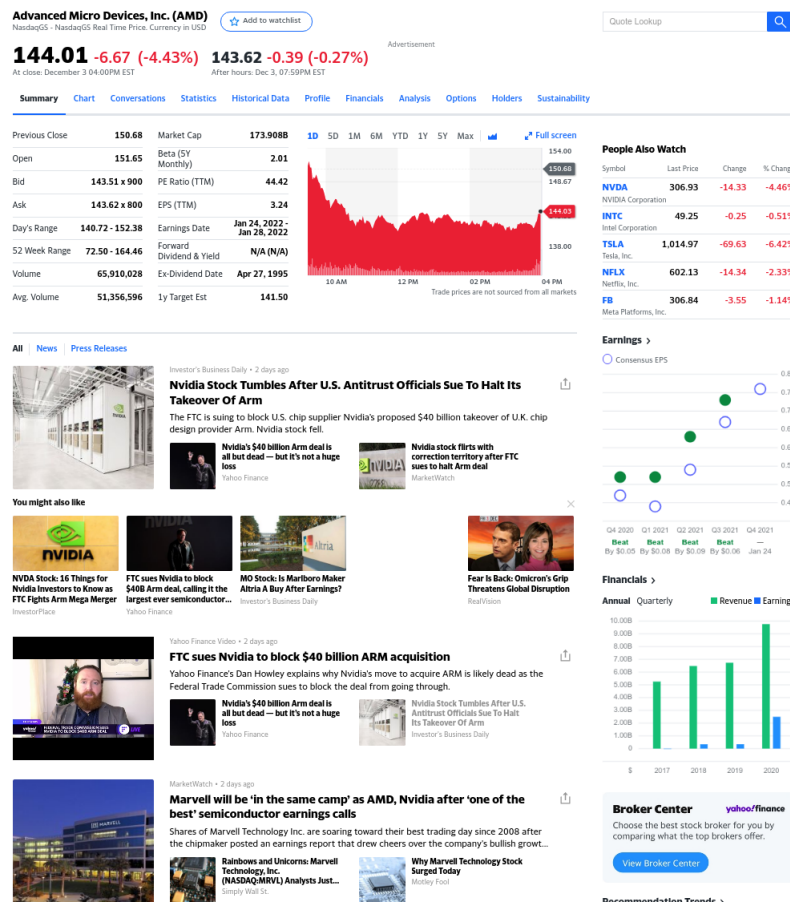


Figure 2.3: Yahoo Finance page for the company Advanced Micro Devices. Related news articles section is visible at the bottom. Each of those articles contains explicit mentioning of AMD as well as a link to its page.

Some finance websites or personal finance app, such as Revolut, outsource their financial data acquisition tasks to companies, such as Tradeweb or Refinitiv, a company that provides global financial market data. Refinitiv produces Eikon [5], a set of software tools called designed for financial professionals to monitor and analyze financial data, including financial news articles. Eikon is highly customizable and is accessible as either as a desktop client application, a Web app, mobile app and as an API for software development purposes. It is not publicly disclosed how Eikon captures financial news data.

Chapter 3

Requirements

3.1 Non-functional requirements

To produce a viable solution to the problem analysis of this dissertation, I chose to implement the experiment with Python programming language. Python is an interpreted general-purpose high-level programming language that is designed to be highly readable and possesses a variety of data analysis and natural language processing (NLP) toolkits and a programming language that I happen to be most fluent in.

For the experiment to be successful and fit into designated time frame, the code needs to be relatively fast and reproducible. Although Python is widely considered to be constrained in the former due to its design philosophy and inclusion global interpreter lock (GIL) mechanism, I overcome this restriction by using the relatively simple and boilerplate code-free built-in and external libraries to parallelize discrete tasks using multiple processes (for CPU bound tasks) or threads (For I/O tasks, such as Web querying) and vectorizing numerical computations on arrays using a popular numerical processing package NumPy [10]. To satisfy the latter, I follow programming practises as described by PEP 8 [29], the official guide for Python coding conventions and best practises, as intended by the creator of the programming language, Guido van Rossum.

To incorporate and train the selected sequence-to-sequence modelling technique - Doc2Vec - I chose gensim [27] package for this task. Gensim provides a fast and reliable implementations for a variety of language modelling and representation learning tasks and doesn't require a graphics processing unit (GPU), which considerably contributes to reproducibility of this experiment. Instead, natural language processing tasks in gensim can be optimized by Cython, a C-like extension and superset of Python programming language.

To provide a rapid and easily interpretable experiment evaluation experience, I design a Web-based graphical user interface (GUI) for my user study following best Web development and user experience (UX) practices, prioritizing minimal eye and cursor travel distance and designing a column-based page layout to emulate the experience of reading an online news website or a newspaper.

3.2 Functional requirements

In this section of the dissertation, I present a list of functional requirements in Table 3.1 modelled after MoSCoW [4] prioritization technique. These requirements describe what the designed system must actually be able to accomplish and will be referenced throughout the dissertation in discussions

where they will be relevant. The requirements are prefixed by Req-[*Category*][*Number*] naming scheme in an attempt to provide an easier interpretation, where *Category* is one of the categories of MoSCoW method: **M** - *Must have*, **S** - *Should have*, **C** - *Could have* and **W** - *Won't have* (this time). In addition to providing a framework for formally declaring requirements, MoSCoW prioritization method dictates that if project delivery timescale runs short, *Should have* and *Could have* requirements should be the discarded first.

ID	Requirements	MoSCoW
Req-M1	The designed solution must provide a ranking of companies that are relevant to a given news article, based on semantic text matching.	Must
Req-M2	The designed solution must generate short text summaries for given news articles.	Must
Req-M3	When producing company rankings, the proposed solution must have an ability to function for cold-start articles and companies (i.e. those not present when training).	Must
Req-M4	The solution must have an associated Web application, which enables users to find relevant news for financial companies.	Must
Req-M5	The Web app needs to serve rankings of related news articles for a specified company.	Must
Req-M6	The Web app should visualise news articles and list related companies to enable exploration of related stocks.	Must
Req-S1	The Web app should provide company search functionality.	Should
Req-S2	The Web app should be adapted to provide a evaluation interface for quantifying the quality of the ranking capabilities of the back-end services.	Should
Req-C1	The designed solution could incorporate state-of-the-art sequence-to-sequence modelling techniques based on transformers such as GPT-3 or BERT.	Could
Req-C2	The experimental setup for testing the designed solution could incorporate a Cranfield-style ranking evaluation protocols [4].	Could
Req-W1	The designed solution won't track stock prices to measure its correlation with a given news article, perform sentiment analysis, nor provide financial advice of any kind.	Won't

Table 3.1: MoSCoW requirements table

Chapter 4

Design & Implementation

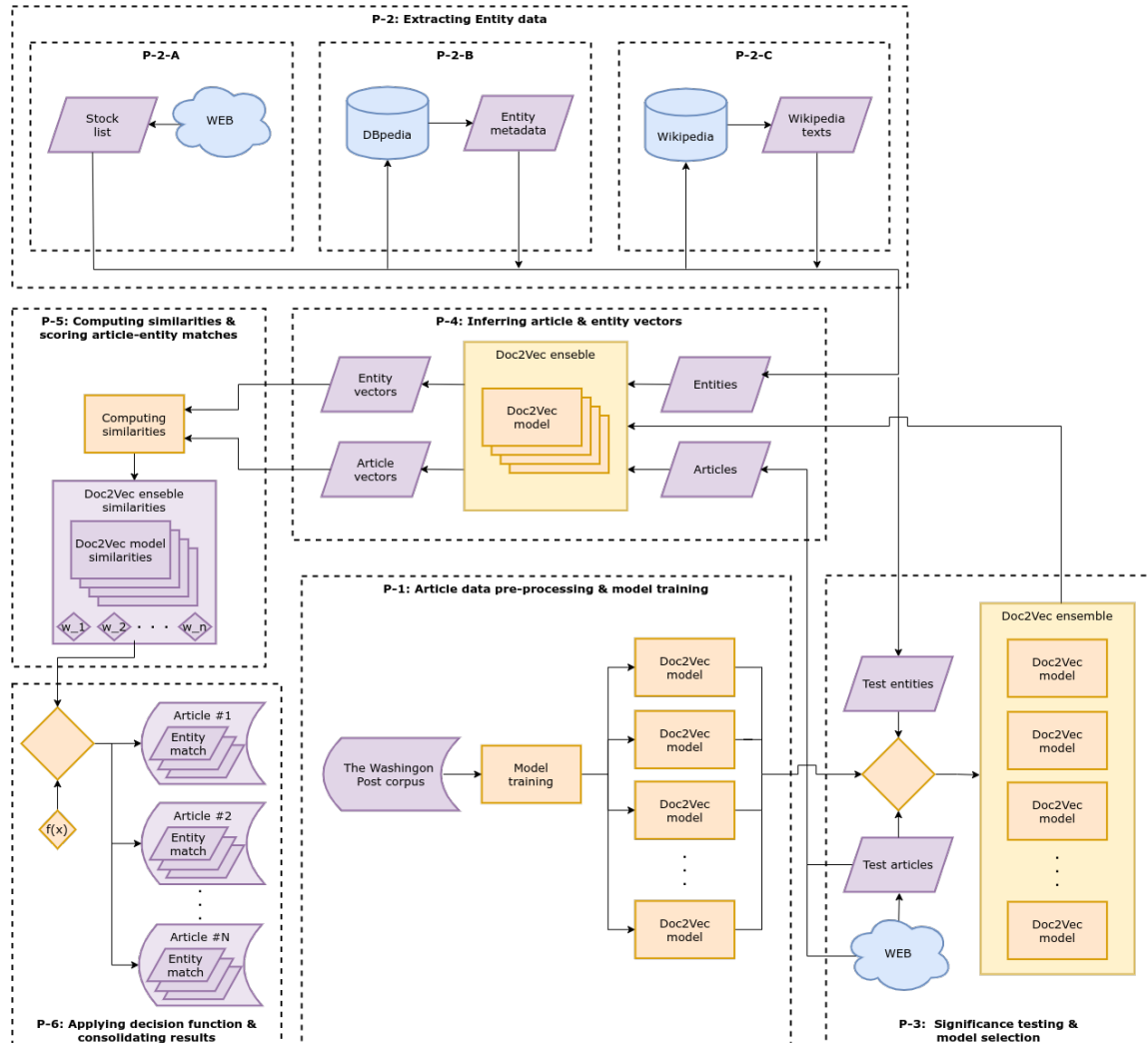


Figure 4.1: Architecture diagram of the proposed solution

4.1 Design considerations

To propose a solution to the problem analysis introduced in Chapter 1, I have to consider approaches that would fit the desired criteria and meet the requirements as described in Table 3.1. First I will consider most common or simplest approaches to natural language processing tasks and discuss their drawbacks and later I shall introduce approaches that would prove to be more advantageous, which I have considered while working on the proposed solution.

Simple approaches

- Companies could be identified using complex regular expressions (*regex*, for short), which could capture mentions of company names or their ticker symbols. However, the former approach would be deficient in that regard that companies are often referenced by their short-hand names, for example Amazon.com, Inc., which is the official name for the company, is often referred simply as Amazon, which is also a river and a rainforest in South America and Tesla, Inc. is simply referred as Tesla, which is named after the famous physicist, Nikola Tesla. Other companies are perhaps more known by their alternate names, such as Petr leo Brasileiro S.A., more commonly known as Petrobras, or Advanced Micro Devices, Inc., almost always referred to simply as AMD. Therefore, searching for company mentions by their name or would require a database for company name variations, which would require a lot of manual work and would constantly need to be updated, should new companies become publicly traded.
- Another approach could perform regex search for company ticker symbols instead. However many companies have very rudimentary ticker symbols, either single letters or formulating some other common English language word, such as Agilent Technologies (traded as “A”), Citigroup Inc. (“C”), Coinbase Global, Inc. (“COIN”), Deutsche Bank AG (“DB”, a short-hand for *database*), ServiceNow, Inc. (“NOW”), AT&T Inc. (“T”) and many more. Therefore it’s extremely possible that using regular expressions to capture ticker symbols and identify associated companies would yield lots of false positives.
- Other common baseline approaches include simple one-hot encodings or bag-of-words models, where models would match word or term frequencies and co-occurrences and document similarities would be scored by measures such as S rensen-Dice coefficient [7] and Jaccard similarity [14]. However, if these approaches are used, all terms that are present in the document are considered equally important, which is most generally considered to be an undesired result. Rare and discriminative terms that would greatly contribute to describing *aboutness* of the document would contribute to similarity scores as equally as non-informative terms, such as English articles, pronouns, adverbs and other words, which are generally considered as *stopwords*.
- More advanced models could be used based on a combination of both term frequency and inverse document frequency (TF-IDF). This approach expands on the idea that rare terms (or n-grams) are more informative than common terms and assigns a TF-IDF score for each term based on some chosen variant for the weights of its two components. The drawback of this approach is that TF-IDF requires that we define our document collection (and effective vocabulary) during model training step, which means that for new documents, unseen, and possibly extremely useful and discriminative terms, such as company names, may be discarded during TF-IDF weight transformations due to them not appearing in our predefined document collection and considerable amount of *aboutness* of the document would be lost.
- Another completely different approach would be to use pre-trained language models with

named entity recognition (NER) parsers. One such industrial-strength natural language processing tool, spaCy [13], is able to parse organizations, locations, names, and more.

Entity label	Description
CARDINAL	Numerals that do not fall under another type.
DATE	Absolute or relative dates or periods.
EVENT	Buildings, airports, highways, bridges, etc.
FAC	Countries, cities, states.
GPE	Any named language.
LANGUAGE	Named documents made into laws.
LAW	Non-GPE locations, mountain ranges, bodies of water.
LOC	Monetary values, including unit.
NORP	Nationalities or religious or political groups.
ORDINAL	"first", "second", etc.
ORG	Companies, agencies, institutions, etc.
PERCENT	Percentage, including. "%"
PERSON	People, including fictional.
PRODUCT	Objects, vehicles, foods, etc. (not services)
QUANTITY	Measurements, as of weight or distance.
TIME	Times smaller than a day.
WORK_OF_ART	Titles of books, songs, etc.

Table 4.1: SpaCy’s named entity recognition parser labels and label descriptions.

However, in practice, spaCy’s NER parser require substantial configuration to work as expected, especially for a task like described in this dissertation, as visible in Table 4.2, where spaCy was used to perform NER task on the article “NIO: Consumers Will Soon Realize NIO Has What Other EVs Are Missing, BaaS”, in which the final chosen approach (introduced later) did considerably better. An excerpt of this article and the its results can be found in Appendix A.

However, all of these approaches are inadequate for this task in multiple ways: either they require manually compiling lists of complex rules and configurations, or they all require the document collection and associated dictionaries/terms/n-grams to be defined during training time. In order to overcome these shortcomings, we have to look elsewhere, to the realm of word/sentence/paragraph embeddings, sequence-to-sequence (seq2seq) learning and deep neural networks [2]. These approaches could be given an input, be it text, image, audio or video, and transform them into sequences of floating point numbers, often interpreted as a set of vectors in linear space, which then could be compared using simple linear algebra techniques, such as cosine similarity.

While (as of this writing) the most popular techniques for such a task involve Transformers [30] (deep neural networks combining encoder and decoder architectures with attention mechanism) they require a substantial amount of prior research to be made in order to be used effectively. Most natural way of venturing on this path is to start with word embeddings, such as Word2Vec [22][23] or GloVe [26].

These word embeddings could be inferred for each word in an article, using either pre-trained models or by training models myself with an appropriate corpus. Resulting vectors could be averaged together to form a central “word” (or, borrowing from k-means clustering terminology, an “exemplar”) vector. Resulting vectors from both the news article and company (however I choose

N-gram	Label	Correct?	Expected result or comment
NIO	ORG	Yes	Refers to both electric vehicle manufacturer NIO Inc. and its ticker symbol.
NIO House Thesis NIO Inc.	ORG	No	The captured n-gram is an amalgamation of paragraph headers and runaway words.
Tesla	PERSON	No	Expected to be recognized as a company.
BYD	ORG	Yes	Electric vehicle manufacturer BYD Motors Inc.
OTCPK	ORG	Yes	Refers to OTC Markets Group Inc. (marketplace for over-the-counter securities) <i>Pink</i> tier.
XPeng	GPE	No	Expected to be recognized as a company.
Li Auto	PERSON	No	Expected to be recognized as a company.
Tesla	ORG	Yes	Electric vehicle manufacturer Tesla Inc.
XPeng	ORG	Yes	Electric vehicle manufacturer XPeng Motors.
the NIO ES6	ORG	No	Expected to be recognized as a product.
V3	PRODUCT	Yes	Refers to Tesla’s V3 Supercharging network
NIO ES8	ORG	No	Expected to be recognized as a product.
EV	PRODUCT	Yes	Acronym for electric vehicle.

Table 4.2: A representative sample of entities that could be useful in detecting article *aboutness* from spaCy’s named entity recognition parser that given the article “NIO: Consumers Will Soon Realize NIO Has What Other EVs Are Missing, BaaS” (see Appendix A)

to represent it) could be compared together either by their distance from each other or the angle between them.

However, there’s a natural alternative - paragraph vectors, or Doc2Vec [17]. Doc2Vec is self-supervised paragraph embedding technique designed to represent full paragraphs and large bodies of text as vectors. In short, during model training step, in addition to training word vectors on either context words prediction task (described as PV-DM method, standing for *paragraph vector - distributed memory*) or center words given context (described as PV-DBOW method, standing for *paragraph vector - distributed bag of words*), Doc2Vec either concatenates or averages additional vector, mean to represent the whole document, with each word window in the document. Despite being considered an older technique, perhaps even inferior to the newer methods, Doc2Vec is known to have outperformed other approaches in Text Retrieval Conference (TREC) tasks in recent years [9]. Given its simplicity and promising results, Doc2Vec is indeed the approach I used in my proposed solution.

To serve rankings of related companies to a given news article, Doc2Vec models must be trained with appropriate data. I chose The Washington Post news corpus (The WaPo news corpus, as it is called) for this task. The Washington Post is one of the most prestigious daily newspapers in The United States, having won over 60 Pulitzer Prizes and famous for its investigative journalism style. The WaPo news corpus contains over 700,000 news articles published from January 2012 through December 2020. Although using this specific training corpus is not a requirement, I consider using it (or a similar news corpus dataset) to be an advantageous as half my test and validation data will consists of news articles as well, with preferably similar journalistic style.

Serving rankings of related companies and Doc2Vec design architecture also requires that I represent companies as texts as well. To accomplish this, I serve Wikipedia article text associated with each company as the text representing the said companies. The advantage of doing that is two-fold: First, doing so means that both test/validation news articles and Wikipedia articles are not present in the training data, thus the model does not need to be retrained should new companies go public, and second, this allows for models to work for cold-start companies, thus satisfying requirement Req-M3. Using Doc2Vec and comparing article-company cosine similarities I could produce a list of related companies, therefore enabling the user to find related companies for a given news article. This list could also be inverted, so that for the user could find related news articles for a given stock (or company).

That concludes the design considerations section and next I shall guide the reader through implementation details of the project.

4.2 Architecture overview

As visible in Figure 4.1, the proposed system consists of six distinct phases that need to be run sequentially. While phases **P-1** and **P-3** can be viewed as the main building block of the experiment and only need to be performed once, the remaining phases compose the final product and should be perpetually updated with new data.

4.3 Implementation details

Raw WaPo corpus is preprocessed. Raw text is extracted from the HTML-like markup structure. Only the most common tags are extracted, such as titles, subtitles, quotes and paragraphs. All embedded external media, such as Tweets and videos, are discarded. These content pieces are then joined together into a single line representing the full article and saved on disk, one line per article, in batches of 10,000 lines.

These batch corpus files are then fed into gensim pre-processing and model training pipeline. Before that, each line is tokenized using standard procedures. Text is lowercased and filtered for only alphanumeric values to remain. Standard nltk [20] library's english stopwords set is applied, which is an important difference to the original Distributed Representations of Sentences and Documents paper [17].

Tokenized text is then transformed into `TaggedLineDocument` format, as preferred by the gensim model training workflow, preserving original article ids, should they ever be needed to be inspected in the raw WaPo corpus dataset again.

A variety of Doc2Vec models were trained, albeit with some hyperparameters that are common to all. As explained in the Chapter 2, all models were trained using distributed memory architecture (PV-DM) and with a concatenation (instead of an average) of paragraph vectors and word vectors

and with negative sampling training algorithm, which was described in one of the two original Word2Vec papers [22].

Other common hyperparameters that were common to all models include minimum word count (number of times a token has to be observed in the vocabulary in order not to be discarded) which was set at five, and two other hyperparameters that were left at their default values: hyperparameter controlling subsampling of the most frequent words was set at $1e-5$ and the eponymous negative sampling distribution exponent was left at $3/4$ (a decision by original researchers which raised some questions [3]) which is used to shape the negative sampling distribution. Both of these parameters were originally chosen heuristically and the decision to leave them at their default values in my proposed solution in combination of discarding stopwords will be reflected upon in Chapter 6 and future work.

Having all of the common hyperparameters explained, Table 4.3 show the combinations of standard Doc2Vec parameters I set for models to be trained with.

Vector dimensions	Window sizes	Negative samples	Epochs
50	1	10	20
75	2	20	40
100	3	30	
150	4		
200	5		
300			

Table 4.3: Principal Doc2Vec model parameters the models were trained with.

P-2: Extracting Entity data

This phase of the project concerns extract, transform and load (ETL) operations on stock metadata, which throughout the text I will refer as Entities. It consists of three sub-phases, getting a list of all available stocks to invest in in a popular personal finance and trading app Revolut, querying DBpedia for additional information about the companies and using Wikipedia API to get associated texts.

P-2-A: Getting a list of available stocks

A list of publicly traded stocks on a mobile personal finance app Revolut is pulled from from the Internet as-is on 5th of July, 2021, containing the data fields of company (or stock) name, ticker symbol, industry and sector. Additionally, a sample of 49 most traded stocks (by volume) during 19th of August, 2021 was recorded in anticipation that these stocks will have more associated relevant news stories, a dataset which will be needed to be constructed around that date to evaluate initially trained Doc2Vec models using Cranfield-style ranking evaluation protocols.

P-2-B: Querying DBpedia for company data

A series of resource description framework (RDF) queries were constructed using stock data fields from a step before in an attempt to retrieve the full company name, a brief description about the company and most importantly, page ID of the associated Wikipedia article. Out of 915 companies in the original dataset, for 780 companies this set of data fields is successfully retrieved and manually validated.

P-2-C: Querying Wikipedia for associated texts

For each company, the associated Wikipedia article is queried using Wikipedia API using two modes - full article text and just the summary (the opening paragraph of the article) in anticipation that the trained Doc2Vec models may not work well with long (or short) texts. Only main text bodies are retrieved (minus the sidebar and paragraph headings). The texts are then immediately pre-processed and tokenized using the same techniques as the training data was during model training step.

This process is repeated for every link in every main company article (which I shall refer to as child entities), except that for child entities only the summary of the article is queried in an anticipation that the steps of inferring entity vectors would consume too much time and would not be viable in production nor development settings.

The process of querying Wikipedia API is parallelized across entities using threads.

P-3: Significance testing & model selection

A variety of 52 articles about the sample of 49 most traded stocks (or entities) during an earlier date are gathered from variety of sources on the Internet, where each entity in the sample is referenced in the sample articles at least once. Sample articles are pre-processed using the same steps as the training data was during the model training phase.

An experiment is set up following Cranfield-style ranking evaluation protocols [4]. Each article (or query) is labelled and a score assigned for most relevant stocks (documents), moderately relevant stocks and irrelevant stocks. These scores are then saved in a query relevance file (qrels).

Due to a random nature of initial vectors in the inferring vectors step, 100 vectors for each article and 10 vectors for each parent and child entities are inferred to mitigate random fluctuations of similarity scores in later steps.

In calculating similarities step, a Cartesian product for each article and entity in the sample is formed, in total a thousand article and entity vector pairs for each article.

A normalized (to range between 0 and 1) cosine similarity metric is defined:

$$NS_C = 1 + \frac{S_C}{2} \quad (4.1)$$

Where:

$$S_C = f(s, k) = \frac{\vec{A} \cdot \vec{E}_{s,k}}{\|\vec{A}\| \|\vec{E}_{s,k}\|} \quad (4.2)$$

Where \vec{A} and $\vec{E}_{s,k}$ are article and entity vectors respectively, s denotes a strategy whether to use full text of the entity Wikipedia article or just a summary and k denotes the kind of entity vectors used, whether parent entity or child. This means s (for “strategy”) becomes the first run parameter that needs to be defined before relevance testing can be done. Note, that for child entities, only the summary of their Wikipedia article is considered.

The final score for an entity in relation to the article is thus given by:

$$Score(s, p) = p * f(s, k = \{parent\}) + (1 - p) * f(s, k = \{child\}) \quad (4.3)$$

Where p is the ratio of parent to child entity normalized cosine similarity scores.

These scores are then computed with vectorized linear algebra operations using popular scientific computing library NumPy and parallelized using processes across all models.

An exhaustive parameter grid of run parameters $s = \{full, summary\}$ and $p = \{0, 0.1, \dots, 0.9, 1\}$ is formed and run files are generated for each initially trained Doc2Vec model and for each run parameter combination.

Run files are evaluated by their normalized discounted cumulative gain (nDCG) [15] measure and significance testing is performed (detailed in section 5.1) to find the best models to form an ensemble for downstream tasks.

P-4: Inferring article & entity vectors

In the previous phase, the selected models for the ensemble performed extremely poorly only on a handful articles, the types of which can be categorized as interview style, product news, articles describing stock market movements or articles analyzing whether a particular stock is a buy (“Due Diligence” or “DD” for short). Some of these articles are removed from the dataset before performing the user study and replaced with unseen, investigative journalism-style validation articles (to combat potential over-fitting issues), the majority of which does not reference a particular stock at all, but perhaps discusses some economic sector or industry.

By now, it’s worth mentioning that neither test nor validation articles were present in the training dataset and models are validated by comparing inferred vectors of both unseen articles and entity texts (which are treated just like other article), thus satisfying requirement Req-M3.

The process that was described previously in **P-3** is in part performed again, but with the full entity dataset and the enhanced article dataset: default values of 100 article vectors and 10 entity vectors are inferred for each model. The code to perform that is parallelized using processes across all models in the ensemble.

P-5: Computing similarities & scoring article-entity matches

The calculating similarities step described in **P-3** is repeated again, this time using full entity dataset and modified article dataset. Similarity scoring process is not changed, but the best performing run parameters ($s = \{full\}$, $p = \{0.8\}$) from Cranfield-style ranking evaluation significance testing step are maintained. The code to perform scoring is parallelized using processes across entities for each model, but each individual article-entity similarity score is persisted to disk in order to save progress should the parallel process needed to restarted at later time.

After computing individual article-entity score, these scores are consolidated into single file for each model. Model weights are chosen heuristically and applied to article-entity scores for each model to give a final averaged article-entity score.

Model	Weight
Doc2Vec(dm-c,d100,n20,w2,mc5,s1e-05,t4,ep40)	1.1
Doc2Vec(dm-c,d75,n20,w1,mc5,s1e-05,t4,ep20)	0.9
Doc2Vec(dm-c,d50,n20,w2,mc5,s1e-05,t4,ep20)	1.2
Doc2Vec(dm-c,d100,n20,w3,mc5,s1e-05,t4,ep40)	0.9
Doc2Vec(dm-c,d100,n20,w4,mc5,s1e-05,t4,ep20)	0.9

Table 4.4: Model weights applied to similarity scores

It’s worth noting that of all chosen models, Doc2Vec (dm-c, d100, n20, w4, mc5, s1e-05, t4, ep20) is the single model that was not significantly better during model evaluation phase, but was included to represent all viable window sizes and was observed to perform well in more problematic articles, as will be discussed in Section 5.1.

P-6: Applying decision function & consolidating results

Having computed all article-entity scores, a decision function needs to be defined to determine which entities will be reported as matches for each article. As expected, almost all article-entity score distributions are highly right-skewed, with most similar entities being on the right tail of the distribution. A cut-off function is defined:

$$t(scores) = \max\{scores\} - \left(-\frac{3}{4} + skewness(scores)^2 + 2\right) * std(scores) \quad (4.4)$$

Where *skewness* is computed with Fisher-Pearson coefficient of skewness [8], *scores* are computer entity similarity scores and *std* is standard scores deviation. This formula was chosen experimentally during model testing phase and was found to work well with test articles. Companies where $score \geq t(scores)$ are reported as matches. If there are more than 10 companies that meet the criteria, only the top 10 are reported, as is with cases where no apparent skewness is observed, meaning the score distribution is approximately normal, which implies the ensemble model did not produce viable matches.

Another important design choice was to “anchor” the threshold to the maximum score value instead of, say, the mean (for example, reporting matches if $score \geq mean(scores) + 2 * std(scores)$). The reasoning behind this is that the former approach guarantees at least one entity match while the latter might not. While “anchoring” the thresholding function to mean would perhaps be more reasonable in production, it would be senseless for users to rate an article with no matches during model validation phase without complete knowledge or experience of stock trading.

Articles with stock matches and their inverse (entities with article rankings) are then persisted to disk and passed on to the Web evaluation app for the user study.

Chapter 5

Evaluation

5.1 Cranfield-style ranking & model evaluation

As discussed in Section 4.3 step **P-3**, a small sample of 49 companies and 52 associated articles were tested and an ensemble of a few best performing Doc2vec models were selected following Cranfield-style ranking evaluation protocols [4]. The protocol consist of the following steps:

Forming query relevance file (qrels)

First, the test articles are labelled and a score is assigned for each relevant entity. For entities that are most relevant (such as directly referenced in the article), a score of 2.0 is assigned, for moderately relevant entities (such as belonging to the same economic sector and industry) a score 1.0 is assigned and for every other entity the score is assigned as zero. Then qrels file is compiled, consisting of the following line format:

```
[article_id] 0 [entity_ticker_symbol] [entity_relevance_score]
```

Where entity relevance score is the score (either 2.0, 1.0 or zero) that is assigned during labelling.

Forming run files

For each candidate model and run parameter configuration a run file is compiled. Each run file consists of lines following the format:

```
[article_id] Q0 [entity_ticker_symbol] [rank] [entity_score] [run_id]
```

Where `entity_score` is computed following procedures described Section 4.3 step **P-3** and rank is simply the order of that score in the sorted list of scores for that document.

Choosing evaluation measure & evaluating run files

I chose normalized discounted cumulative gain (nDCG) [15] as a measure for evaluating run file ranking quality, as I believed mean average precision (MAP) not to be as informative metric for ranking entities (although I do report it) as the usefulness, or *gain* of the entity based on its position in the result list, which nDCG measures.

Each run file is evaluated against each article defined in qrels and their nDCG scores are averaged. Run files are sorted by their average nDCG and pair-wise significance testing is performed.

Significance testing and model selection

For N runs evaluated in the previous step, $N-1$ pairwise significance tests (measuring paired differences of nDCG across all sample articles), all against the leading run (by its average nDCG). By the time of writing this dissertation (as more models were trained and their runs were evaluated throughout development process for scientific curiosity purposes) the only runs that were not significantly different (using two-tailed t-student tests with significance level of 5%) than the leading runs are described in Table 5.1.

Vector size	Negative samples	Window size	Epochs	Avg. nDCG
100	20	1	40	0.815255
100	20	2	40	0.806648
100	20	1	20	0.805743
100	20	2	20	0.797485
75	20	2	20	0.794338
50	20	1	20	0.794188
100	20	3	40	0.792668
75	20	1	20	0.788650
50	20	2	20	0.780151

Table 5.1: Runs that were not significantly different from the leading run by average nDCG score. All runs had run parameters set at $p=\{0.8\}$ and $strategy=\{“Full”\}$.

It is worth noting that only two-sided significance testing was performed with no corrections such as Bonaferrini’s [12]. As such no more significance testing were performed for fear of obtaining false positive results. The leading run by its average nDCG score was chosen to be manually inspected for its performance on individual test articles.

However, after examining the (at that time) leading run’s performance on the test articles, the results were less than satisfactory. Problematic articles were identified and Cranfield-style ranking evaluation experiment was performed again, only this time considering only the list of problematic articles and keeping a tally of run appearances in the top 5 runs for each problematic article. The final ensemble model is chosen, such that most effective, but slightly different vector, window and negative sample sizes are represented to avoid overfitting. The final chosen ensemble of models can be seen in Table 4.4.

5.2 User study & model validation

User study consisted of two tasks, which will be referred as Task One and Task Two. The evaluation Web app received 838 total responses, 461 responses for Task One and 377 responses for Task Two.

In Task One, study participants are given 46 different articles to be evaluated, written in variety of journalistic styles, some containing discussions about particular stocks, some discussing a particular economic sector or industry and others are written in investigative journalism style and don’t discuss any particular stocks at all. These articles are paired with a list of stocks (ranging from 1 to the maximum of 10) that the ensemble model determined to be matches.

Study participants are asked evaluate the quality of the results using five-point Likert scale [18] suited for measuring quality (*Very Poor*, *Poor*, *Acceptable*, *Good*, *Very Good*) based on the following criteria:

Bezos' Blue Origin loses NASA lawsuit over SpaceX \$2.9 billion lunar lander contract

Bezos' Blue Origin loses NASA lawsuit over SpaceX \$2.9 billion lunar lander contract. KEY POINTS. 2. Federal Judge Richard Hertling sided with the defense in his ruling, completing a months-long battle.

Ratings: 10

AT&T CFO Pascal Desroches Updates Shareholders

AT&T CFO Pascal Desroches Updates Shareholders. DALLAS, September 15, 2021—(BUSINESS WIRE)—Pascal Desroches, senior executive vice president and chief financial officer of AT&T Inc.* (NYSE:TT), spoke today at the Bank

Ratings: 10

Ford to build \$11.4 billion mega campuses for electric car production

Ford to build \$11.4 billion mega campuses for electric car production. Saying Ford is driving head on into a future of electric vehicles may be an understatement. Think more along the line of hauling a—, Stanton,

Ratings: 10

Oil recovers from U.S. inflation-driven plunge

Summary. 2. Biden administration could order oil reserves release. Oil recovers from U.S. inflation-driven

NIO: Consumers Will Soon Realize NIO Has What Other EVs Are Missing, BaaS

Summary

1. NIO is a leading player in the Chinese electric vehicle market and their unique technology grants it several long-term advantages over competitors.
2. Their battery as a service technology solves a lot of short and long-term problems with EVs and cars in general.
3. China has been on a legislative tear recently, hurting many domestic companies. However, NIO's mission lines up with many Chinese goals and is likely to enjoy preferential treatment.
4. The current fall of Chinese stocks makes NIO an attractive long-term investment.

NIO logo and the Nio's user center, NIO House

Thesis

NIO Inc. (NIO) is one of the leading electric vehicle producers in China, the largest market for EVs. While still being outsold by Tesla (TSLA), the fast-growing nature of the company and market offers promise. Their battery as a service subscription option offers NIO a unique advantage and solves many of the problems consumers have with EVs. The stock has seen some weakness over the past three months, mainly because of China's massive regulatory crackdown. NIO, however, will likely see preferential treatment as the goals of the company align with that of the Chinese Communist Party.

NIO's current position

To understand NIO's position in the Chinese EV market, we first need to understand the Chinese EV market.

Model results

rank	index	stock_name	ticker_symbol	sector	industry	comment
1	763	Xpeng	XPEV	Consumer Durables	Motor Vehicles	Xpeng or Xiaopeng Motors (Chinese: 小鹏汽车), also known as XMotors.ai, is a Chinese electric vehicle manufacturer. The company is headquartered in Guangzhou, with offices in Mountain View, California in the US and is publicly traded on the New York Stock Exchange.
2	502	NIO	NIO	Consumer Durables	Motor Vehicles	NIO (Chinese: 蔚来; pinyin: Wèilái) is a Chinese multinational automobile manufacturer headquartered in Shanghai, specializing in designing and developing electric vehicles. The company is also involved in the FIA Formula E Championship, although NIO has ended funding the team after selling to Lisheng Racing.
3	430	Li Auto Inc.	LI	Producer Manufacturing	Auto Parts: OEM	Li Auto Inc., also known as Li Xiang, is a Chinese electric vehicle manufacturer headquartered in Beijing, with manufacturing facilities in Chanzhou.

Model evaluation:

Very Poor
Poor
Acceptable
Good
Very Good
Submit

Article metadata

title NIO: Consumers Will Soon Realize NIO Has What Other EVs Are Missing, BaaS

company NIO

url <https://seekingalpha.com/article/4457311-nio-has-what-other-evs-are-missing-baas>

Evaluation instructions: Task One

You are asked to evaluate model performance on a 'stock retrieval' task for a given article.

The model outputs a list of stocks from a pre-defined list based on their 'relatedness' to a given article. These could be interpreted as companies that are related to the topic of the article, to the topic company (if there is one), as competitors to each other or as related to each other in some other way. Task One is to evaluate the validity and coherence of such lists.

The 'rank' column of model results reflect the 'relatedness' of the company to the article and can be considered when evaluating model results.

You should use the following evaluation criteria in your evaluations as well as fill optional checklist below.

- Do model results reflect the economy sector discussed in the article?
- Do model results reflect the economy industry discussed in the article?

Reasons for evaluation (optional):

☐ Topic company is/is not present in model results

☐ Model results contains companies that are/are not is similar economic sector

☐ Model results contains companies that are/are not is similar economic industry

☐ Model results contain companies that are/are not related to the topic company

☐ Personal knowledge on the topic company and/or on the topic of the article

Figure 5.1: Task One evaluation Web app interface for article “NIO: Consumers Will Soon Realize NIO Has What Other EVs Are Missing, BaaS” as seen on a Full HD resolution desktop.

- Do model results reflect the economy sector discussed in the article?
- Do model results reflect the economy industry discussed in the article?
- Is the topic company (or companies), if there are some, present in model results?
- Are companies (if there are some) that are related to the topic company in some way present in model results?
- Personal knowledge on the topic of the article/sector/industry/topic company(ies)/related companies at hand.

Study participants are also invited to provide optional reasons for their evaluation, which for Task One are the following:

- Topic company is/is not present in model results.
- Model results contains companies that are/are not is similar economic sector.
- Model results contains companies that are/are not is similar economic industry.
- Model results contain companies that are/are not related to the topic company.
- Personal knowledge on the topic company and/or on the topic of the article.
- Other reasons (unspecified).

Options for providing optional reasons combine both positive and negative sentiment in order not to clutter the Web app UI and to save vertical space. It is expected that the true intended sentiment will be inferred from the associated evaluation label (positive sentiment for labels *Good* and *Very Good* and negative sentiment for *Very Poor* and *Poor*), although this approach makes true sentiment difficult to discern for evaluations that are *Acceptable*. This task, in accordance to the specified criteria, aim to evaluate company ranking effectiveness, validity and coherence. In short, study

participants are asked to consider if these ranking lists contain companies that are not similar to each other in the sense of their economic sector or industry unless they have personal knowledge that claims otherwise.

For Task Two, the model results of Task One are inverted, such that instead of evaluating company rankings for a given article, study participants are asked to evaluate rankings of associated articles for a given company. Instead of providing a full list of 780 stocks in the stock dataset, I chose a representative sample of 75 companies such that each combination of economic sector and industry is represented by at least a single company, with a caveat that the said company must have at least one associated article, except for companies that are directly mentioned in some of the articles, which are included automatically. This is done to distinguish any discrepancies between evaluations for different economic sectors or industries and detect if there is any bias in the model towards companies of particular type. Sector-industry combinations that are represented and those that are omitted in Task Two are shown in Table 5.2.

Sector	Industry
Communications	Major Telecommunications
Consumer Durables	Electronics/Appliances
Consumer Durables	Motor Vehicles
Consumer Durables	Recreational Products
Consumer Non-Durables	Apparel/Footwear
Consumer Non-Durables	Beverages: Alcoholic
Consumer Non-Durables	Beverages: Non-Alcoholic
Consumer Non-Durables	Food: Major Diversified
Consumer Services	Hotels/Resorts/Cruise lines
Consumer Services	Movies/Entertainment
Consumer Services	Other Consumer Services
Electronic Technology	Aerospace & Defense
Electronic Technology	Computer Peripherals
Electronic Technology	Electronic Components
Electronic Technology	Semiconductors
Electronic Technology	Telecommunications Equipment
Energy Minerals	Integrated Oil
Energy Minerals	Oil & Gas Production
Finance	Finance/Rental/Leasing
Finance	Financial Conglomerates
Finance	Investment Banks/Brokers
Finance	Major Banks
Finance	Regional Banks
Health Technology	Biotechnology
Health Technology	Pharmaceuticals: Major
Industrial Services	Oil & Gas Pipelines
Industrial Services	Oilfield Services/Equipment
Non-Energy Minerals	Other Metals/Minerals
Non-Energy Minerals	Steel
Process Industries	Chemicals: Major Diversified
Producer Manufacturing	Auto Parts: OEM
Retail Trade	Department Stores
Retail Trade	Internet Retail
Technology Services	Internet Software/Services
Technology Services	Packaged Software
Transportation	Airlines
Utilities	Electric Utilities

(a) Economic sector and industry combinations that are represented in the 75 entity subsample.

Sector	Industry
Commercial Services	Advertising/Marketing Services
Commercial Services	Financial Publishing/Services
Communications	Specialty Telecommunications
Communications	Telecommunications Equipment
Communications	Wireless Telecommunications
Consumer Durables	Homebuilding
Consumer Non-Durables	Household/Personal Care
Consumer Services	Broadcasting
Consumer Services	Cable/Satellite TV
Consumer Services	Restaurants
Electronic Technology	Computer Communications
Electronic Technology	Computer Processing Hardware
Electronic Technology	Electronic Equipment/Instruments
Energy Minerals	Oil Refining/Marketing
Finance	Investment Managers
Finance	Life/Health Insurance
Finance	Real Estate Development
Finance	Real Estate Investment Trusts
Health Technology	Medical Specialties
Health Technology	Pharmaceuticals: Generic
Industrial Services	Environmental Services
Non-Energy Minerals	Aluminum
Non-Energy Minerals	Precious Metals
Process Industries	Chemicals: Specialty
Process Industries	Containers/Packaging
Process Industries	Industrial Specialties
Producer Manufacturing	Electrical Products
Producer Manufacturing	Miscellaneous Manufacturing
Retail Trade	Apparel/Footwear Retail
Retail Trade	Drugstore Chains
Retail Trade	Electronics/Appliance Stores
Retail Trade	Specialty Stores
Technology Services	Data Processing Services
Technology Services	Information Technology Services
Transportation	Railroads
Utilities	Alternative Power Generation
Utilities	Gas Distributors

(b) Economic sector and industry combinations that are *not* represented in the 75 entity subsample.

Table 5.2: Economic sector and industry combinations in the stock dataset.

For Task Two, the provided evaluation criteria are:

- Are articles in the *Relevant* column truly relevant to the stock in question?
- Are articles that are directly related the stock in question present in the *Relevant* column?
- Do articles in the *Relevant* reflect the sector and industry of the stock in question?
- Personal knowledge on the stock and/or sector and/or industry.

Activation	ATVI	Consumer Durables	Recreational Products
Blizzard	SPCE	Electronic Technology	Aerospace & Defense
National Oilwell Varco	NOV	Industrial Services	Oilfield Services/Equipment
Kraft Heinz	KHC	Consumer Non-Durables	Food: Major Diversified
Element Solutions	ESI	Process Industries	Chemicals: Major Diversified
China Mobile	CHL	Communications	Telecommunications Equipment
Coty	COTY	Consumer Non-Durables	Household/Personal Care
FuelCell Energy	FCEL	Electronic Technology	Electronic Equipment/Instrument
Marathon Petroleum	MPC	Energy Minerals	Oil Refining/Marketing
KKR & Co	KKR	Finance	Investment Managers
Inovio Pharmaceuticals	INO	Health Technology	Medical Specialties
Iamgold Corp	IAG	Non-Energy Minerals	Precious Metals
Ralph Lauren	RL	Retail Trade	Apparel/Footwear Retail
SS&C Technologies Holdings	SSNC	Technology Services	Information Technology Services
Brookfield Infrastructure Partners	BIP	Utilities	Alternative Power Generation
Sempra Energy	SRE	Utilities	Gas Distributors
LI Auto Inc.	LI	Producer Manufacturing	Auto Parts: OEM
NIO	NIO	Consumer Durables	Motor Vehicles

Relevant articles to the stock

NIO: Consumers Will Soon Realize NIO Has What Other EVs Are Missing, BaoS
Rank: 1
Their battery as a service technology solves a lot of short and long-term problems with EVs and cars in general. NIO is a leading player in the Chinese electric vehicle market and their unique technology grants it several long-term advantages over competitors. China has been on a legislative tear recently, hurting many domestic companies. However, NIO's mission lines up with many Chinese goals and is likely to enjoy preferential treatment.

EV Competition Has Arrived: Does Tesla Have The Answer?
Rank: 2
Tesla trades at a high 20x P/S and is overvalued. Summary: EV Competition Has Arrived: Does Tesla Have The Answer?

Fresno County Will Soon Be Home to the Largest Green Hydrogen Plant on the West Coast
Rank: 3
Fresno County Will Soon Be Home to the Largest Green Hydrogen Plant on the West Coast. The project also includes the construction of a new wastewater treatment facility that will supply the water needed by the plant and provide recycled water to Mendota, an increasingly water-stressed agricultural hub in Fresno County. The California facility will expand Plug Power's network of plants that already includes sites in New York, Tennessee and Georgia. Combined, they will supply 500 tons of liquid green hydrogen daily by 2025, preventing 4.3 million metric tons of carbon dioxide emissions from entering the atmosphere.

Irrelevant articles to the stock

AT&T CFO Pascal Desroches Updates Shareholders
AT&T CFO Pascal Desroches Updates Shareholders. DALLAS, September 15, 2021--(BUSINESS WIRE)--Pascal Desroches, senior executive vice president and chief financial officer of AT&T Inc. (NYSE:TT), spoke today at the Bank of America Media, Communications & Entertainment Conference, where he provided an update to shareholders. AT&T also continues to experience healthy demand for HBO Max in both domestic and international markets. The company recently announced its plans to expand into six European countries next month with plans to launch in at least 14 additional territories in Europe in 2022. As previously indicated, AT&T expects most of the subscriber growth in the second half of the year to come from outside the United States due to the strategic decision to cease offering HBO Max as a subscription on Amazon Channels. The company anticipates this decision will likely impact total HBO Max / HBO domestic subscribers and net additions in the third quarter. However, Desroches reiterated that AT&T's guidance for 70 million to 73 million global HBO Max and HBO subscribers by the end of 2021 factors in the impact of this decision.

Bezos' Blue Origin loses NASA lawsuit over SpaceX \$2.9 billion lunar lander contract
Bezos' Blue Origin loses NASA lawsuit over SpaceX \$2.9 billion lunar lander contract. KEY POINTS. 2. Federal Judge Richard Hertling sided with the defense in his ruling, completing a months-long battle.

Ford to build \$11.4 billion mega campuses for electric car production
Ford to build \$11.4 billion mega campuses for electric car production. Saying Ford is driving head on into a future of electric vehicles may be an understatement. Think more along the line of hauling a-- Stanton, Tennessee will see the construction of Blue Oval City, which Ford says will aid in building an "expanded" electric-truck lineup. Currently, Ford has released the all-electric SUV called the Mach-E and soon will begin producing the all-electric F-150 Lightning. The 3,600-acre campus will cover nearly 6 square miles, at a total of \$5.6 billion.

Stock metadata

Stock name	NIO
Ticker symbol	NIO
Sector	Consumer Durables
Industry	Motor Vehicles
Comment	NIO (Chinese: 蔚来; pinyin: Wèilái) is a Chinese multinational automobile manufacturer headquartered in Shanghai, specializing in designing and developing electric vehicles. The company is also involved in the FIA Formula E Championship, although NIO has ended funding the team after selling to Lisheng Racing.

Evaluation instructions: Task Two
You are asked to evaluate model performance on an "article ranking" task for a given stock. Model results are inverted and for each stock the model outputs a list of "relevant" articles (those in which this specific stock was present in model results) and "irrelevant" articles (those which did not). Task Two is to evaluate the accuracy, ranking and coherence of such lists.
You can use stock metadata table (see: above) to learn about the stock and click on a news article to read it in full.
An additional "Not Sure" evaluation rating is added in case there are no apparent cases whether the article should be (or not) included in either of the lists and/or the evaluator is not familiar with the topic enough to answer.
You should use the following evaluation criteria in your evaluations as well as fill optional checklist below.

Reasons for evaluation (optional):

- ☐ Are articles in the "Relevant" column truly relevant to the stock in question?
- ☐ Are articles that are directly about the stock in question in the "Relevant" column?
- ☐ Do articles in the "Relevant" reflect the sector and industry of the stock in question?

Reasons for evaluation (optional):

- ☐ Irrelevant articles in the relevant column
- ☐ Relevant articles in the irrelevant column
- ☐ Topics of articles in the relevant column are not about stock sector or industry
- ☐ Clearly related articles to the stock in the irrelevant column (evaluation criteria no. 2)
- ☐ Personal knowledge on the topic of the stock and/or on the topic of the article

Model evaluation:

☐ Not Sure
 ☐ Very Poor
 ☐ Poor
 ☐ Acceptable
 ☐ Good
 ☐ Very Good

Figure 5.2: Task Two evaluation Web app interface for the company NIO as seen on a Full HD resolution desktop.

For Task Two, there are two important differences that are worth mentioning. First, an additional *Not Sure* evaluation rating was added. This is because for some of the companies in Task Two dataset, an assumption is made it is difficult to discern what the company actually does without investing considerable amount of research and therefore it would be difficult evaluate their article ranking effectiveness. However, they are still included in Task Two dataset as they may represent some economic sector and industry combination which would otherwise be unrepresented. Preliminary exploratory data analysis showed, that such ratings were rare. The other important difference is that study participants are asked not to consider articles in the *Irrelevant* column unless they are specifically related to the company at hand. The reasoning behind this is that the company at hand may be indeed relevant to some of the articles in the *Irrelevant* column, but not so much as to make it to the top of its rankings.

The experiment evaluation Web app was hosted on a popular hosting platform Heroku and the evaluation platform will be able to be accessed for the duration of this dissertation evaluation (should the reader chose to do so) by following provided links, [Task One](#) and [Task Two](#).

User study evaluation results

As can be seen on Figure 5.3, although the overall result across both task seems promising, the proposed solution performed significantly better on Task One than Task Two. I suspect there are few reasons why this might have happened:

1. As stated before, articles are presented with a list of associated companies. While overall the

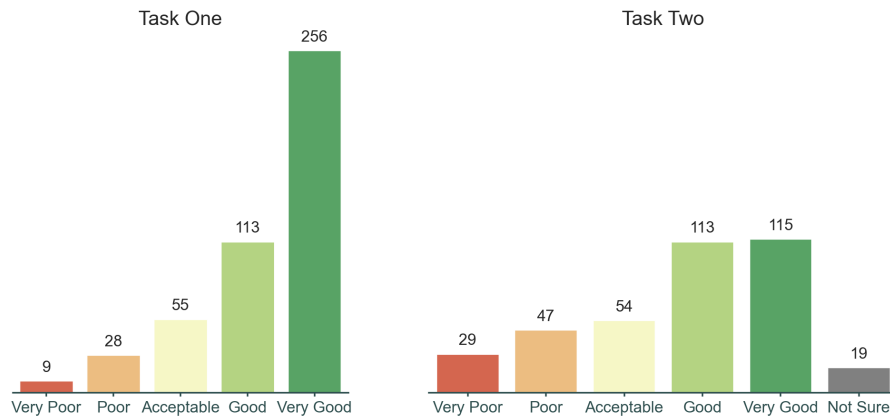


Figure 5.3: User study evaluation results for Task One and Task Two. Numbers on the top of the bars represent the number of associated responses the task received.

list seems coherent and valid and receive either a *Good* or *Very Good* response (notice the omission of a label such as *Excellent*), there might be companies here and there that are not related to the article. When this list is reversed in Task Two and companies are presented with article rankings instead, *Relevant* column contains the irrelevant article, hence the given ratings express less positive sentiment.

2. Expanding on the first point, the cut-off function described in Equation 4.4 was chosen heuristically and might have been too permissive and allowed for unrelated entities to be included as matches, which again, when reversed during Task Two, provided unrelated articles in its ranked list.
3. Articles in Task Two are ranked by the same score entity received in Task One. For example:

Company Advanced Micro Devices received a score of 0.693045 for article “AMD CEO Sees Chip Shortage Easing in 2022” in Task One. It was the top score of all other companies listed in the article as matches. Therefore, this article might have received a *Good* or *Very Good* evaluation rating. However, the same company received scores of 0.742016 and 0.697524 for articles “New Benchmark Leak Reveals Impressive Performance Of Intel’s 12900K, 12700K And 12600K Alder Lake Processors” and “Why Nvidia Investors Are Nervous Today” respectively. Since the same scores are used to rank articles in Task Two, article “AMD CEO Sees Chip Shortage Easing in 2022” is ranked third although Advanced Micro Devices is the principal subject of the article. Therefore, evaluation ratings for this company in Task Two might have been less positive.

The principal way to overcome the problem described in Point 3 would have been to apply min-max scaling for entity matches in Task One, which was not done. This will be referenced as a point in Future Work section, as it probably affected the evaluation ratings for Task Two considerably. Next we will look at what optional reasons were provided for Task One evaluations in Figure 5.4.

Looking at Figure 5.4 we can clearly see the trend that for for just about every rating study participants care equally whether the results contain the topic company at hand or other entities that are in the similar sector and industry. It would also seem to have been sufficient to provide a single option for both economic sector and industry.

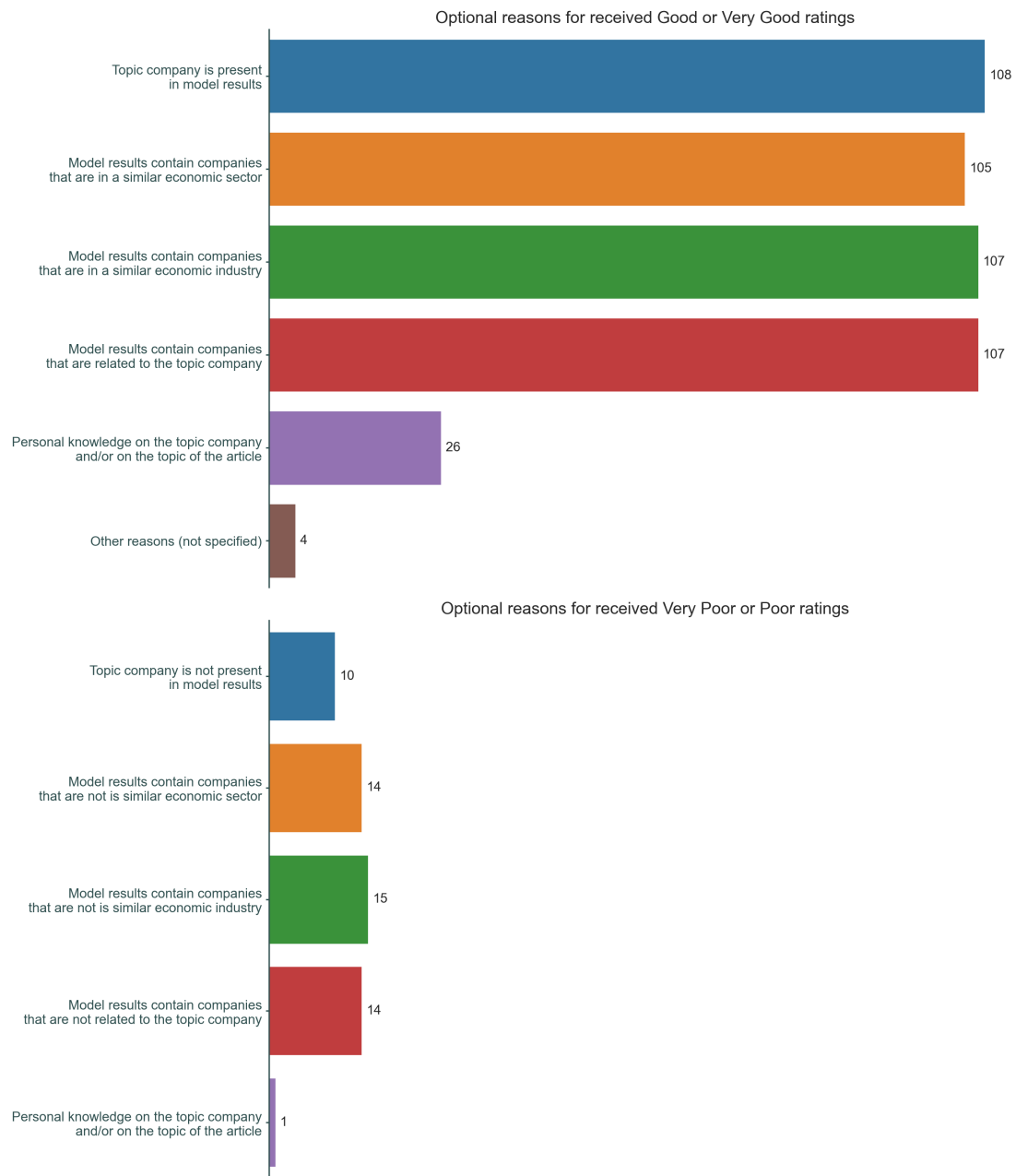


Figure 5.4: Optional reasons provided for Task One evaluations.

For Task Two, as can be seen in Figure 5.5, although user study participants were asked not to strongly consider seemingly related articles in the *Irrelevant* column (as those entities might have not made in the top of the rankings), they clearly expressed their opinion that some of those articles were indeed relevant. Participants were also dissatisfied with irrelevant articles in the *Relevant* column, but as have been explained earlier, this was due to noise among the signal in entity rankings in Task One.

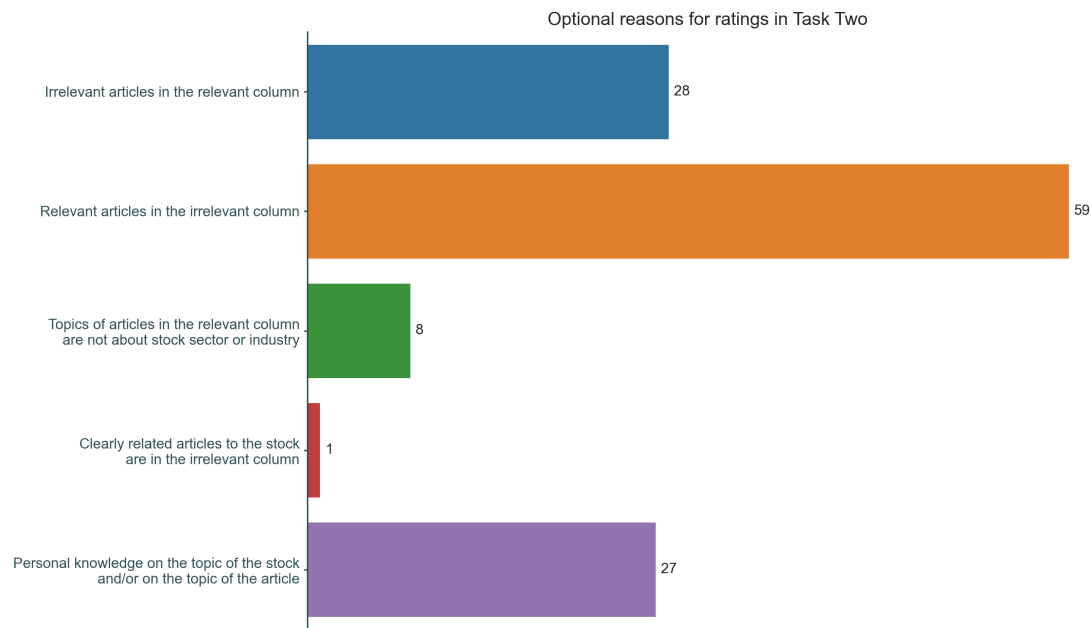


Figure 5.5: Optional reasons given for ratings in Task Two.

It also might be interesting to see what kind of entities received a *Not Sure* rating in Task Two in Table 5.3. Although the sample size for *Not Sure* evaluation is small, a small faint can be seen that the user study participants were least familiar with a combined Energy/Non-Energy Minerals industry.

Company	Ticker symbol	Industry	Sector
Roku	ROKU	Consumer Durables	Electronics/Appliances
Coty	COTY	Consumer Non-Durables	Household/Personal Care
Keurig Dr Pepper	KDP	Consumer Non-Durables	Beverages: Non-Alcoholic
Carnival	CCL	Consumer Services	Hotels/Resorts/Cruise lines
New Oriental Education & Tech Group	EDU	Consumer Services	Other Consumer Services
FuelCell Energy	FCEL	Electronic Technology	Electronic Equipment/Instruments
Chevron	CVX	Energy Minerals	Integrated Oil
Exxon Mobil	XOM	Energy Minerals	Integrated Oil
Banco Santander (Brasil)	BSBR	Finance	Regional Banks
KKR & Co	KKR	Finance	Investment Managers
Pfizer	PFE	Health Technology	Pharmaceuticals: Major
Rio Tinto	RIO	Non-Energy Minerals	Other Metals/Minerals
Iamgold Corp	IAG	Non-Energy Minerals	Precious Metals
Freeport	FCX	Non-Energy Minerals	Other Metals/Minerals
United States Steel	X	Non-Energy Minerals	Steel
Element Solutions	ESI	Process Industries	Chemicals: Major Diversified
SS&C Technologies Holdings	SSNC	Technology Services	Information Technology Services
YY	YY	Technology Services	Internet Software/Services

Table 5.3: Companies that received a *Not Sure* evaluation rating in Task Two

5.3 Requirements Validation

In this section, having discussed all of the design choices and implementation details, I will refer the reader back to the MoSCoW requirements table (Table 3.1) and take a look at the requirements again to discuss how they were fulfilled (if they were fulfilled) or justify if they could not have been met. As the requirement Req-W1 is a *Won't have* requirement, it is omitted as it is implicitly met.

Requirement	Comment	Completed?
Req-M1	The provided solution produces rankings based on a Doc2Vec model and is tested and validated as described in Sections 4.3 and 5.1.	Yes
Req-M2	External solution - PreSumm [19] (GitHub project repository here) - that is not part of the source code of this project was used to generate short text summaries.	No
Req-M3	The provided solution treats entities just like other unseen articles and does not require them to be present in the training dataset, as discussed in Section 4.1.	Yes
Req-M4	The provided solution includes a Web application as seen in Figures 5.1 and 5.2 and mentioned Section 4.1. The experiment associated with the Web app is described in full in Section 5.2.	Yes
Req-M5	The Web application, as part of the evaluation experiment (which referred in this dissertation as Task Two), serves news rankings for a representative subset of 75 financial companies, which are sorted by their similarity to said articles as computed by a custom scoring function and described in full in Section 4.3.	Yes
Req-M6	The Web application, as part of the evaluation experiment (which referred in this dissertation as Task One), provides a list of 46 online publications of written in various journalistic styles for which rankings of related companies are available to be evaluated, as visible in Figure 5.2.	Yes

Table 5.4: MoSCoW *Must have* requirements validation.

Requirement	Comment	Completed?
Req-S1	As discussed in Section 5.2 a representative sample of 75 companies (among the total 780 in the full dataset) were selected for user study evaluation. This is because I considered getting evaluations for each of the companies in the full dataset unrealistic and user study participants might have instead favoured to evaluate companies they knew better, making the study less impartial. Instead, the companies that were selected in such a way, that they represent as many different combinations of the economic sector and industry as possible, with a caveat that they must have at least one associated article. Given a relatively small size of resulting sample and the GUI design choices for the Web app, I felt that companies are easily accessible without a search functionality by way of scrolling, as can be visible in Figure 5.1.	No
Req-S2	Although best-worst scaling [21] is found to be more reliable than numerical rating scales [16], I chose a five-point Likert scale [18] suited for surveys measuring quality for evaluating article and stock matches (<i>Very Poor, Poor, Acceptable, Good, Very Good</i>), as the remaining timescale at that point dissertation writing couldn't have accommodated training additional Doc2Vec ensemble models for best-worst comparisons.	Yes
Req-C1	Although laconically titled <i>Attention Is All You Need</i> [30] is a seminal paper and probably one of the most influential natural language processing papers in recent years, in order to understand the Transformer architecture (with attention mechanism) and its popular implementations such as BERT [6] required researching a significant amount prior art to use them effectively, which unfortunately did not fit in the remaining timescale (at that point) of this dissertation.	No
Req-C2	The model testing experimental set-up, introduced in Section 4.3 and expanded upon in Section 5.1, does indeed follow Cranfield-style ranking evaluation protocols	Yes

Table 5.5: MoSCoW *Should have* and *Could have* requirements validation.

Chapter 6

Conclusion

In this dissertation, I defined my main research question - is automatic inferral of entities from various online publications possible based on semantic text matching, reviewed related academic work, gathered and declared formal and informal requirements, formulated the design and experimental set-up, implemented the proposed solution and evaluated the resulting system based on both automatic and user study techniques. I think I can with confidence conclude that the answer to my research question is indeed possible even with an older set of tools (than the current state-of-the-art techniques) and as user study suggest, the results are positive and promising. It still remains to be seen what can be done with the current state-of-the-art techniques, which will be part of my future work, which I shall describe next.

6.1 Future work

During the development for this project, I made an assumption based on prior research that Doc2vec models following PV-DM architecture trained with negative sampling algorithm would be the best approach if Doc2vec approach was chosen at all, despite the fact that the original researchers recommended combining PV-DM and PV-DBOW architectures. For future work, I would like to repeat the experiment using several Doc2vec model ensembles combining both architectures.

Expanding on the first point, one of the *Could have* MoSCoW requirements, Req-C1 was not met, as an assumption was made that in order to use Transformer architecture effectively a lot of prior art would need to be studied for which ample time was not available. For future work I would like to challenge this notion and propose additional solutions with state-of-the-art Transformer architectures.

For evaluation scheme I used a five-point Likert scale suited for measuring quality. Such a rating scale is proven to be less reliable than other rating schemes, such as Best-Worst rating scheme [16]. Expanding on the first two points, if multiple solutions were provided for user study instead of a single one, I would like to repeat the experiment by comparing the multiple proposed solutions using Best-Worst rating scheme.

Finally, the results for Task Two during user study were less desirable than for Task One. This was due to noise among the signal (irrelevant companies among the relevant ones) and better entity ranking, filtering or thresholding procedures needed to be researched and applied, rather than the currently used ones, chosen heuristically.

Appendix A

NIO: Consumers Will Soon Realize NIO Has What Other EVs Are Missing, BaaS

In this appendix, I will present an article “NIO: Consumers Will Soon Realize NIO Has What Other EVs Are Missing, BaaS” as an example of an article that the models generally did well with as judged by both myself and user study participants. Because the original source of this article, website [Seeking Alpha](#), hides articles behind a paywall after a limited amount of reads, I will present a short excerpt of the article here and link the original source at the end, followed by a short discussion on model results.

A.1 Article excerpt

Sep. 27, 2021 3:21 PM ET.

Summary

- NIO is a leading player in the Chinese electric vehicle market and their unique technology grants it several long-term advantages over competitors.
- Their battery as a service technology solves a lot of short and long-term problems with EVs and cars in general.
- China has been on a legislative tear recently, hurting many domestic companies. However, NIO’s mission lines up with many Chinese goals and is likely to enjoy preferential treatment.
- The current fall of Chinese stocks makes NIO an attractive long-term investment.

Thesis

NIO Inc. (NIO) is one of the leading electric vehicle producers in China, the largest market for EVs. While still being outsold by Tesla (TSLA), the fast-growing nature of the company and market offers promise. Their battery as a service subscription option offers NIO a unique advantage and solves many of the problems consumers have with EVs. The stock has seen some weakness over the past three months, mainly because of China’s massive regulatory crackdown. NIO, however, will likely see preferential treatment as the goals of the company align with that of the Chinese Communist Party.

NIO's current position

To understand NIO's position in the Chinese EV market, we first need to understand the Chinese EV market.

First off, ICE vehicles are still dominant in China, but the market for EVs seems to be in the early stages of exponential growth and is expected to continue to grow rapidly, with EVs expected to make up 34% of total Chinese car sales in 2025.

Being an early leader and domestic producer of EVs should position NIO well for this type of growth.

The Chinese electric vehicle market is pretty fragmented, but Tesla has a solid lead. Chinese companies BYD (OTCPK:BYDDY), NIO, XPeng (XPEV), and Li Auto (LI) are also relevant. In 2020, Tesla sold 135k vehicles in China while BYD shipped 130k units, NIO shipped 44k units, XPeng shipped 27k units, and Li shipped 15k.

It is worth noting that these figures will likely be quite different for the full year 2021. The market is growing and changing rapidly, and new EVs like the Wuling Hong Guang MINI EV made by SAIC-GM-Wuling, a sub \$5,000 car with a top speed of 62 mph, are flying off the shelves.

NIO's advantage

I firmly believe that NIO's battery as a service and battery swap offerings will benefit NIO in the long run. This is rooted in the belief that efficient battery swapping can solve or alleviate many of the pain points that consumers have with EVs, but consumers have not realized that yet.

Battery swap program

NIO offers a monthly subscription to drivers that grants them access to their 484 swap stations. NIO claims to have completed over 2 million swaps. This unique approach, previously abandoned by Tesla after a poor performance in 2015, offers more advantages to their consumers than immediately apparent. Battery as a service (BaaS) solves or alleviates a lot of problems that all EVs have.

This article was originally written by author [Moonshot Equity Analysis](#) and the original text can be accessed [here](#) [Accessed November 29th, 2021].

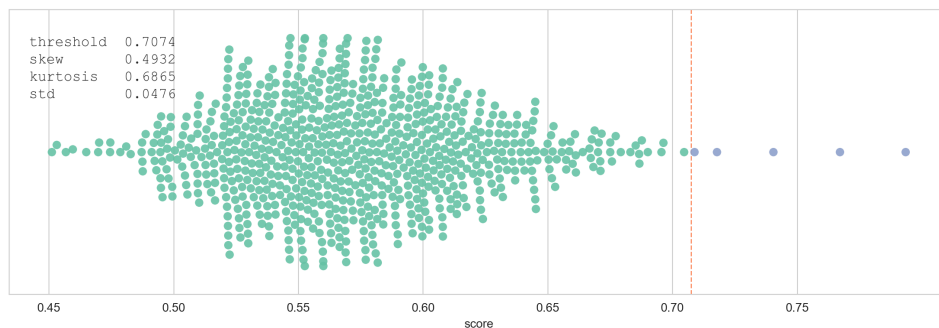


Figure A.1: Swarmplot of companies associated with the article in question. Companies are represented as dots in the plot: turquoise dots are companies that scored too low to be considered as matches, grey dots are companies that surpassed the threshold, which is visible as the orange dotted line. Some of the score distribution parameters are visible in the upper left portion of the figure.

Model results						
rank	index	stock_name	ticker_symbol	sector	industry	comment
1	763	Xpeng	XPEV	Consumer Durables	Motor Vehicles	Xpeng or Xiaopeng Motors (Chinese: 小鹏汽车), also known as XMotors.ai, is a Chinese electric vehicle manufacturer. The company is headquartered in Guangzhou, with offices in Mountain View, California in the US and is publicly traded on the New York Stock Exchange.
2	502	NIO	NIO	Consumer Durables	Motor Vehicles	NIO (Chinese: 蔚来; pinyin: Wèilái) is a Chinese multinational automobile manufacturer headquartered in Shanghai, specializing in designing and developing electric vehicles. The company is also involved in the FIA Formula E Championship, although NIO has ended funding the team after selling to Lisheng Racing.
3	430	Li Auto Inc.	LI	Producer Manufacturing	Auto Parts: OEM	Li Auto Inc., also known as Li Xiang, is a Chinese electric vehicle manufacturer headquartered in Beijing, with manufacturing facilities in Changzhou.
4	566	Proterra	PTRA	Finance	Financial Conglomerates	Proterra is an American automotive and energy storage company based in Burlingame, California. The company designs and manufactures electric transit buses and electric charging systems. Proterra's Catalyst series includes transit buses ranging from 35 feet (11 m) to 40 feet (12 m) in length and various battery configurations. Buses are charged through an overhead charging station that is placed at maintenance facilities as well as route terminals. Founded in 2004, it became a public company in June 2021 through a business transaction with a Nasdaq-listed special-purpose acquisition company.
5	676	Tesla	TSLA	Consumer Durables	Motor Vehicles	Tesla, Inc. is an American electric vehicle, clean energy, and technology company based in Palo Alto, California, United States. Tesla designs and manufactures electric cars, battery energy storage from home to grid-scale, solar panels and solar roof tiles, and related products and services. In 2020, Tesla had the most sales of battery electric vehicles and plug-in electric vehicles, capturing 16% of the plug-in market (which includes plug-in hybrids) and 23% of the battery-electric (purely electric) market. Through its subsidiary Tesla Energy, the company develops and is a major installer of photovoltaic systems in the United States. Tesla Energy is also one of the largest global suppliers of battery energy storage systems, with 3 gigawatt-hours (GWh) installed in 2020.

Figure A.2: Entity ranking results as outputted by the ensemble model.

A.2 User study evaluation

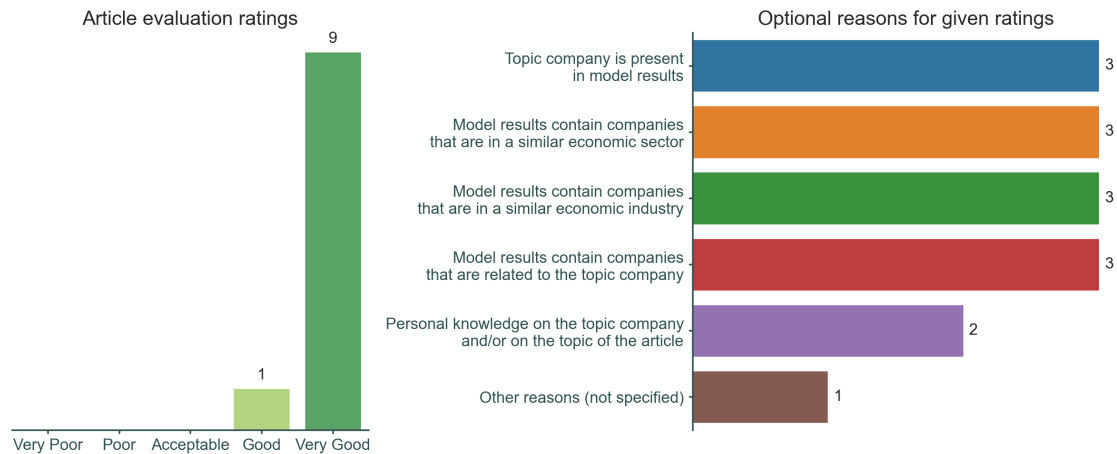


Figure A.3: User study evaluation results. (Left) main article evaluations as provided by five-point Likert scale. (Right) Optional reasons supplied for evaluation ratings. As can be seen from both Figure A.2 and Figure A.3, user study participants were extremely satisfied with this particular article results.

Bibliography

- [1] Yoshua Bengio, Holger Schwenk, Jean-Sébastien Senécal, Frédéric Morin, and Jean-Luc Gauvain. *Neural Probabilistic Language Models*, pages 137–186. Springer Berlin Heidelberg, Berlin, Heidelberg, 2006.
- [2] A. Burkov. *The Hundred-Page Machine Learning Book*. Andriy Burkov, 2019.
- [3] Hugo Caselles-Dupré, Florian Lesaint, and Jimena Royo-Letelier. Word2vec applied to recommendation: Hyperparameters matter. *CoRR*, abs/1804.04212, 2018.
- [4] D. Clegg and R. Barker. *CASE Method Fast-track: A RAD Approach*. CASE method. Addison-Wesley Publishing Company, 1994.
- [5] Thomson Reuters Corporation. Thomson reuters eikon quick start guide. [Thomson Reuters Eikon Quick Start Guide](#). Accessed: 2021-12-01.
- [6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. In Jill Burstein, Christy Doran, and Thamar Solorio, editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics, 2019.
- [7] Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.
- [8] David Doane and Lori Seward. Measuring skewness: A forgotten statistic? *J. Stat. Educ.*, 19, 07 2011.
- [9] Kenan Fayoumi and Reyhan Yeniterzi. Ozu-nlp at trec news 2019: Entity ranking. In *TREC*, 2019.
- [10] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Pícus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Hal-dane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.
- [11] Zellig Harris. Distributional structure. *Word*, 10(2-3):146–162, 1954.
- [12] Winston Haynes. *Bonferroni Correction*, pages 154–154. Springer New York, New York, NY, 2013.

- [13] Matthew Honnibal and Mark Johnson. An improved non-monotonic transition system for dependency parsing. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1373–1378, Lisbon, Portugal, September 2015. Association for Computational Linguistics.
- [14] Paul Jaccard. The distribution of the flora in the alpine zone.1. *New Phytologist*, 11(2):37–50, 1912.
- [15] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of ir techniques. *ACM Transactions on Information Systems*, 20:2002, 2002.
- [16] Svetlana Kiritchenko and Saif M. Mohammad. Best-worst scaling more reliable than rating scales: A case study on sentiment intensity annotation. *CoRR*, abs/1712.01765, 2017.
- [17] Quoc V. Le and Tomas Mikolov. Distributed representations of sentences and documents, 2014.
- [18] R. Likert. *A Technique for the Measurement of Attitudes*. Number nos. 136-165 in *A Technique for the Measurement of Attitudes*. Archives of Psychology, 1932.
- [19] Yang Liu and Mirella Lapata. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*, 2019.
- [20] Edward Loper and Steven Bird. Nltk: The natural language toolkit. *CoRR*, cs.CL/0205028, 2002.
- [21] Jordan J. Louviere, Terry N. Flynn, and A. A. J. Marley. *Best-Worst Scaling: Theory, Methods and Applications*. Cambridge University Press, 2015.
- [22] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.
- [23] Tomáš Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. *CoRR*, abs/1310.4546, 2013.
- [24] Tomáš Mikolov, Stefan Kombrink, Lukáš Burget, Jan Černocký, and Sanjeev Khudanpur. Extensions of recurrent neural network language model. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5528–5531, 2011.
- [25] Andriy Mnih and Yee Whye Teh. A fast and simple algorithm for training neural probabilistic language models, 2012.
- [26] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. *EMNLP*, 14:1532–1543, 01 2014.
- [27] Radim Rehurek and Petr Sojka. Gensim–python framework for vector space modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2), 2011.
- [28] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. Learning Representations by Back-propagating Errors. *Nature*, 323(6088):533–536, 1986.
- [29] Guido van Rossum, Barry Warsaw, and Nick Coghlan. Style guide for Python code. PEP 8, The Python Steering Council, 2001.
- [30] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg,

S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.