2020

Word Embedding Driven Concept Detection in Philosophical Corpora

Dylan Hayton-Ruffner

Word Embedding Driven Concept Detection in Philosophical Corpora

An Honors Paper for the Department of Computer Science
By Dylan Hayton-Ruffner

Bowdoin College, 2020

BOWDOIN COLLEGE

COMPUTER SCIENCE

Word Embedding Driven Concept Detection in Philosophical Corpora

Author:
Dylan HAYTON-RUFFNER, *hrdylan@gmail.com*

Advisors:
Fernando NASCIMENTO
Stephen MAJERCIK
Stacy DOORE

May 14, 2020

Bowdoin College, 2020

Contents

***Abstract*—During the course of research, scholars often explore large textual databases for segments of text relevant to their conceptual analyses. This study proposes, develops and evaluates two algorithms for automated concept detection in theoretical corpora: ACS and WMD Retrieval. Both novel algorithms are compared to key word search, using a test set from the Digital Ricoeur corpus tagged by scholarly experts. WMD Retrieval outperforms key word search on the concept detection task. Thus, WMD Retrieval is a promising tool for concept detection and information retrieval systems focused on theoretical corpora.**

I. Statement of Problem

A. Introduction

With the proliferation of the web and digitization, textual data is increasingly accessible on an unprecedented scale. Whether through compiled collections like Wikipedia or traditional web scraping, assembling large textual data sets is a relatively trivial task. However, while digitization has increased our access to information it has not necessarily increased our understanding of it. Data on such a scale is impossible to analyze with human means - reading and writing by hand. Thus, textual databases represent an important opportunity for Natural Language Processing (NLP) research and experimentation. Computational analysis is the only practical method for processing, traversing, and analyzing such large collections.

B. Information Retrieval (IR)

Information Retrieval (IR) is a classic problem in NLP made all the more relevant by the proliferation of text data. At its core, text-based IR helps users deal with the the scale of text data by locating, analyzing, and retrieving documents [22]. IR is relevant to tools like Google, Bing, JSTOR, and Bowdoin OneSearch, which help human agents navigate huge stores of information. In general, IR involves a corpus of text documents of a large, definite size, and a user interested in retrieving information from that corpus. The user communicates their information needs in the form of a query — typically a set of words — that indicates the documents they are interested in. The task of the IR system is to process the query, search the corpus, and return relevant documents to the user. In practice, perfect recall, returning 100% of relevant documents, is difficult. Thus, IR systems must often balance recall, the number of relevant documents returned out of the total set of relevant documents, and precision, the number of relevant documents out of the total number of documents returned in the query. High recall is useless if precision is too low and vice versa [21].

C. Concept Detection

Concept detection, a special case of information retrieval, is the process of finding and retrieving documents that define and expand upon a given concept. Concept detection is typically conducted using theoretical corpora from fields like philosophy, psychology, and literature which structure texts around sets of concepts.

The International Organization for Standardization (ISO) provides a rigorous definition of a concept. An object, as defined by the ISO, is “anything perceivable or conceivable” [29]. Objects have characteristics, which are abstractions of the properties of an object [29]. Concepts combine these characteristics into units of knowledge [29]. For example, the concept ‘planet’, combines all the characteristics of a planet – round, massive, stellar etc. – into a single identifiable entity. The set of characteristics a concept combines is called its intension. Concepts may also be abstract. The concept ‘justice’ combines the terms ‘truth’, ‘right’, and ‘law’ into an idea of judicial equality.

An extension of a concept is the totality of objects to which a concept corresponds [29]. Extensions of ‘planet’ might be Saturn, Jupiter or Earth, but also might include generic objects like ‘heavenly body’ or ‘astronomical body’. A concept can be visualized as a cloud of these extensions, semantically related by the characteristics the concepts contains. The concept relates and describes each object, conveying their characteristics in a single unit.

Just as in IR, in concept detection a user expresses their information need through a query: a word or set of words that refer to a specific concept within a corpus of theoretical texts. The goal of the system is to return segments to the user that are relevant to the definition of the concept. The quality rather than the quantity of the results returned by the system is paramount. The complexities of theoretical corpora prevent users from processing large volumes of information quickly. Theoretical corpora are also often restricted by fair-use copyright law. Databases are allowed to display only a small subset of the entire corpus in response to a user’s query for period of 80 years after the author’s death. Thus, search algorithms have to make a trade off between the number of results shown and the amount of context displayed around each result. Context is indispensable for a researcher, as it frames the meaning and content of each result. Thus, concept detection requires that results be in context. To fulfill this constraint and comply with copyright law, concept detection queries must return small numbers of high quality segments.

Concept detection is vital when conducting a conceptual analysis, in which a researcher explores the expression of a concept in the works of one or more writers [11]. Such studies can cover decades worth of material and require large amounts of time to complete. Concept detection expedites this process by locating all the areas of target texts that are relevant to and useful for the researcher [10].

D. Project Summary

The purpose of this project is to explore, develop, and evaluate word embedding driven concept detection algorithms for use within a philosophical corpus. Word embeddings, high-quality vector representations of words, are able to capture complex semantic relationships in natural language. Thus, word embeddings are a promising computational tool for concept detection, which requires that algorithms traverse nuanced webs of words and concepts. The goal of this project is to develop an algorithm that effectively leverages word embeddings and evaluate it against key word search - a common matching algorithm used by researchers to identify concepts in text.

E. Corpus for Project

One of the key obstacles in concept detection research in philosophy is access to data. In philosophy, many important works either remain in print form or are not organized into systemic and accessible digital collections. However, the Digital Ricouer Project, which digitizes and collects the works of philosopher Paul Ricoeur into an online database, has amassed a large digital corpora of philosophical writings, providing a unique opportunity for text analysis projects. The corpus for this project, drawn from Digital Ricoeur, spans Ricouer's career and consists of 59 French works, 3,466,624 tokens, and 80,545 unique words. The size of this corpus is expected to grow as new texts are digitized and added to the database. While digitized collections exist in both French and English, the french corpus was selected to minimize the effect of human translation, allowing the project to leverage the writer's exact wording.

II. Contribution

This project builds upon the body of IR work discussed in Section IV, by exploring IR techniques in theoretical corpora i.e. the humanities. It is vital to note the importance of IR in theoretical corpora. Improved retrieval systems have the potential to greatly increase the speed of scholarly work by expediting information searches. Online databases like JSTOR and even library search catalogs would also benefit from improvements in IR techniques. The findings of this study are directly applicable to retrieval-based text analysis projects working with theoretical corpora.

This project also expands the breadth of IR research, by working outside of the typical corpora. The TREC data sets, which are the standard in IR research, consist of news corpora. MED is another common data set that contains medical abstracts [5]. Much of IR research has thus been confined to similar, standardized corpora.

There are good reasons for this. First, it allows for a linear narrative in research progress. New techniques can be show to improve upon past iterations in relation to standard metrics. Moreover, these test sets contain text types that are commonly searched and widely in demand - news media, papers, web documents, etc. Thus, new research is judged on how applicable it is to solving the most in demand IR problems, resulting in higher commercial applicability and impact.

However, the focus on standardized data sets limits the applicability of research by narrowing testing to specific types of written works. It is insufficient to assume that the research thus far is directly applicable to theoretical corpora, which differs significantly from medical abstracts, news articles, and web documents. Scientific papers focus on making quantitative and empirical observations using precise language. News documents and media are inherently descriptive, with an emphasis on the clear transmission of information to the reader. On the other hand, theoretical texts are structured as sets of interrelated concepts that are nuanced and obscure by nature. Thus, algorithms that perform well on typical IR data sets do not necessarily generalize to theoretical corpora.

Finally, theoretical corpora are often protected by copyright law. Any retrieval operation on databases with protected material is limited in the amount of information it can return. For instance, the Digital Ricouer Portal is prohibited from showing users more than a small percentage of the text in its databases. Currently, Digital

Ricouer and similar projects solve this issue by returning small portions of the text around any segment that might be relevant to the query. Thus, more information can be returned, even though it lacks both context and specificity. However, a robust concept detection tool has the potential to give retrieval systems greater confidence in the relevance of retrieved segments. A smaller number of documents could be returned with more context, providing meaningful, accurate results.

III. A Test Set for Concept Detection

In order to evaluate a concept detection tool, a test set of segments, tagged by scholarly experts for their relevance to a set of concepts, is required. With this set, the ability of the tool to accurately find and retrieve segments associated with a concept can be assessed against ground truth values backed by scholarly consensus.

Our test set contains paragraphs from two chapters, written in French, from *The Symbolism of Evil* (SM) and *Oneself as Another* (SA). *The Symbolism of Evil* is a mono-graphic work that explores several well defined concepts: 'myth', 'symbol', 'evil'. *Oneself as Another* is an edited collection of Ricoeur's lectures and deals with a multiplicity of concepts. These two works span the breadth of Ricoeur's philosophical career, the former published in 1967 and the latter published in 1992. Each paragraph in the set is tagged with one of four categorical variables - Defines, Relates to, Sub concept, Not related - indicating its relevance to a set of concepts. Segments from SM were tagged for the concepts mythe (myth), homme (man), and symbole (symbol), while segments from SA were tagged for the concepts morale (morale), justice, and sagesse pratique (practical wisdom). These tags indicate categorical judgements of the segment's relationship to the concept. The tags were provided by four scholarly experts on SM and three on SA. The percent agreement among experts was 42.8% for SM and 48.5% for SA.

A. From Categorical Tags to Binary Relevance Judgements

The raw test set consists of segments tagged by experts for each concept. However, these tags represent categorical judgements of the relationship between segment and concept. To evaluate a concept detection algorithm, each segment must be determined to be either 'Relevant' or 'Irrelevant' to the concept detection query. To obtain this set of binary relevance tags, each categorical tag was mapped to one of two relevance tags: 'Relevant' and 'Irrelevant'. The tags 'Defines' and 'Relates to' were mapped to 'Relevant' because they indicate that a segment is either defining or expanding upon the concept. The tags 'Sub concept' and 'Not Related' were mapped to 'Irrelevant'. For the criteria shown to experts for each tag see Appendix A. After mapping the categorical tags to binary relevance tags, the percent agreement among experts was 63.5% for SM and 56.1% for SA.

B. Determining Consensus

Because unanimous agreement was uncommon in the mapped test set, the expert's binary relevance tags had to be aggregated into a final relevance judgement for each segment. A majority-rules algorithm was used to determine this consensus among the expert's tags. The tag reaching a simple majority was adopted as the final relevance judgement. In the event of a tie, the segment was labeled 'Irrelevant' to the concept. For SM, 18% of consensus determinations ended in a ties. Since there were only three judges for SA, no consensus determinations ended in a tie.

IV. Prior Work

A. Boolean Retrieval

The classic solution to IR problems is Boolean Retrieval. In this approach, queries are represented as sets of words interspersed with Boolean modifiers [23]. For example, a query "Bowdoin AND Bear", would match all documents containing both the word "Bowdoin" and the word "Bear". The query "Bowdoin OR Bear" would match any document with either word. This approach has a couple of significant flaws. First, the number of documents returned is very unpredictable. In general, either very little or nearly all of the corpus match the strict criteria of a boolean sequence. Second, when a large number of documents match the query there is no way to rank the results [24].

To solve these difficulties, several new algorithms were proposed in the 1980s. Called Extended Boolean Retrieval, these approaches assign a score to documents, typically in the range (0 - 1), where 0 indicates no match and 1 indicates a complete match [23]. The most successful of these algorithms is p-norm. In this approach, the terms of each document or query are given a weight using TF-IDF, a weighting heuristic that measures the importance of a term to a document. TF-IDF incorporates term frequency, or the number of times a word occurs in a document, and inverse-document-frequency, or the frequency of the word in the corpus. TF-IDF increases with TF and decreases with IDF. Thus, rare words that occur frequently are given higher weights than common words that occur frequently [27]. With TF-IDF as weights, *p*-norm calculates the similarity between a document and a query using the terms and term weights of both, essentially treating the query as if it were a document. The hyper parameter p $(1 < p < \infty)$, changes the behavior of the heuristic. At p = 1, the heuristic behaves like an inner product between the vector of the query term weights and the document term weights. At $p = \infty$, the heuristic behaves like a strict boolean expression. For details on the math behind *p*-norm, see [24].

B. Synonymy and Polysemy

The fundamental issues with Boolean Retrieval, or any approach that involves term matching, are synonymy, many words can mean the same thing, and polysemy, a single word can mean many things [5]. Consider the case when a query term q matches a document term d. Because words have many meanings, q may have an entirely different meaning from d even though they are lexically identical. Consider the case when q does not match d. Because many words mean the same thing, q may mean the same thing as d despite their lexical differences.

For example, a user is querying a database looking for content related to presidential speeches. They formulate a query with the words [PRESIDENT, SPEECH]. Consider the following documents:

1) The president developed a speech impediment over the course of his time in office.
2) Obama gave a good lecture at the beginning of the summit.

Document 1 has multiple terms that match the query: “president” and “speech”. In the case of “president”, the terms both lexically and semantically match. Both the query and the document refer to a president of the United States. The term “speech” in the document and query lexically match but differ semantically. The user references a formal address while the document refers to the act of speaking. Although Document 2 has no lexical matches, it has multiple semantic matches. “President” and “Obama” both refer to US Presidents. “Speech” and “lecture” both reference a formal address. Thus, despite the lexical similarities between the query and Document 1, Document 2 is more relevant. There have been attempts to augment Boolean Retrieval to take into account polysemy and synonymy. Coarse techniques like automatic term expansion and thesaurus construction have both been proposed, but require human oversight and thus are not robust or scalable.

Synonymy and polysemy are especially relevant in concept detection. Because concepts in the humanities are nebulous, an author may discuss a concept without ever using the words most frequently associated with it. For instance, Darwin discusses evolution in *On the Origin Species* without ever using the word evolution. Thus, what computational concept detection systems require is a model capable of coherently and accurately computing the semantic relationships between words. The following sections examine such approaches.

C. Distributional Hypothesis

The distributional hypothesis is a fundamental assumption of computational semantics and underlies many of the techniques describe in the following sections. The distributional hypothesis is best summarized by linguist John Rupert Firth’s famous quote, “you shall know a word by the company it keeps” [1]. The distributional hypothesis asserts that syntactic relationships reflect semantic relationships. Words that have similar syntax, appearing in the text surrounded by similar words, have a similar meaning. With this assumption, the following models are able to leverage statistics, linear algebra, and machine learning to go beyond classical retrieval and infer the semantic structure of natural language.

D. Vector Space Model (VSM)

The Vector Space Model expresses queries and documents in vector space and utilizes vector similarity operations to calculate document relevance to a query. Documents and queries are expressed as vectors by converting the corpus into a co-occurrence matrix. The co-occurrence matrix is a $v \times d$ matrix where v is the size of the corpus’s vocabulary and d is the number of distinct documents in the corpus. Each entry in the matrix, $C_{i,j}$, represents the number of times word i appears in document j. In practical applications, raw term-frequency is often replaced with TF-IDF. For example consider the following corpus:

1) The cow is black.
2) The fish is big.

3) The bear is yellow.

For this corpus, the co-occurrence matrix would be:

	Doc 1	Doc 2	Doc 3
the	1	1	1
is	1	1	1
fish	0	1	0
cow	1	0	0
bear	0	0	1
black	1	0	0
big	0	1	0
yellow	0	0	1

This representation, projects each document into a vector space defined by a basis $B = (t_1, t_2...t_n)$ where each t_i is a term in the corpus' vocabulary [26]. Each column of the co-occurrence matrix gives the vector for a document in this vector space. Notice, that the representation can also be reversed. Words can be represented as vectors in a vector space spanned by the documents.

To determine the distance between a document and a query, the query is represented as a vector in the space $Q = (q_1, q_2...q_n)$ where q_i is the weight of term i in the query, just as if it were a document in the co-occurrence matrix [26]. The similarity between the query and document vectors is calculated using a variety of metrics including, euclidean distance, dot product, and cosine similarity [23].

However, the fundamental assumption of VSMs — that a document's semantic content can be accurately represented in a vector space described by a basis of terms — is flawed. In order to be a basis, the set of terms must be linearly independent, each representing an independent dimension of the space. Thus, the weight of one term in the document is assumed to have no effect on the contribution of other terms in the document. This is not true of natural language. Consider the document: “Obama gave a good lecture at the beginning of the summit”. The terms “lecture” and “Obama” indicate that “summit” refers to a meeting of world leaders and not a mountain peak. Thus, the whole document describes a presidential speech given at a conference rather than on top of a mountain. One dimension of the space, affects the other.

E. Latent Semantic Indexing (LSI)

As proposed by Deerwester, Dumanis and Harshman, Latent Semantic Indexing is a fascinating linear algebra driven approach to retrieval that provides modest improvements over both classical term-matching and vector space models (VSMs) [5]. Like VSM, LSI construcst a vector space in which both documents and words can be represented as vectors. Document-word similarity within this vector space is estimated using distance metrics like cosine similarity. While, basic VSM's use the co-occurrence matrix to form their vector space, LSI takes a more complicated approach. Assuming the the co-occurrence matrix contains noisy data, LSI attempts to find a simplified vector space, called the *latent semantic space*, that approximates the data in the co-occurrence matrix. Instead of representing documents in terms of words, LSI computes a smaller set of latent variables and uses this set to define a basis for it's vector space. Singular Value Decomposition is the primary mechanism for finding the latent space. The co-occurrence matrix is decomposed using SVD, insignificant values are removed from the singular value matrix and the resulting decomposition can be used to estimate the similarity of words to documents, documents to documents, and words to words [2]. For example, the co-occurrence matrix C is decomposed, via SVD, into two orthonormal matrices, T and D, and one diagonal matrix S.

$$C = TSD \tag{1}$$

T and D contain the left and right singular vectors of C and S contains the singular values of C. Insignificant singular values are removed from S to form S_o and their corresponding columns are removed from T and D to form T_o and D_o. The number of values removed is a hyper-parameter. The results is a new matrix, represented by the new simplified decomposition.

$$C_o = T_o S_o D_o \tag{2}$$

Their product can be shown to be the closest approximation of the original co-occurrence matrix by any matrix of the product's rank. This new matrix represents documents and words in a simplified latent semantic space and can be used to calculate document-document, word-word, and document-word similarity. To rank documents according to a given query, the query is represented as a document and the distance between each document and the query is calculated using the simplified matrix. For more details on the mathematics see [5].

With LSI, Deerwester, Dumanis and Harshman, achieved a modest increase in performance over both term-matching and VSMs on standardized data sets. They concluded that “LSI should be regarded as a potential component of a retrieval system, rather than a complete retrieval system" [2].

F. Statistical Language Modeling (LM)

Statistical Language Modeling applies probabilistic models to textual data [17]. In general, these mechanisms are concerned with predicting the probability of a set or sequence of words:

$$P(w_1....w_n) \tag{3}$$

This probability is traditionally estimated using a Markov assumption:

$$P(w_1....w_n) = \sum_{i=1}^{n} P(w_i|w_1...w_{i-1}) \tag{4}$$

These traditional models are called n-gram models [18]. There are many methods of building language models for textual corpora, see [16] for a comprehensive review.

Query likelihood retrieval (QL), proposed by Ponte and Croft, is the basic schematic for applying language modeling to IR. In QL, a language model is estimated for each document in the corpus. Documents are then ranked according to a given query based on the likelihood of the query given each document's language model. Documents that assign high probabilities to the query are said to be relevant to the query [16]. There are many variations of QL retrieval. State-of-the-art iterations of LM in IR include KL-divergence retrieval [20] and the Relevance Model [19].

G. Probabilistic Latent Semantic Analysis: PLSI and LDA

Probabilistic Latent Semantic Analysis applies statistics to the task of Latent Semantic Analysis. Just like LSI, PLSA algorithms assume that co-occurrence data is noisy and conditioned by a set of latent variables. However, where LSI employed SVD to compute these relationships, PLSA uses statistics, leveraging the following generative model. When a writer sits down to create a work they first select a set of latent variables called topics. Each of these topics is itself a bag of words. As the author writes, they select one of their chosen topics, pull out a word, and add it to their document [28]. PLSA algorithms reverse this process, looking at each occurrence of a word in a document and estimating both the mixture of topics the author used to create the document and the mixture of words in each topic. PLSA results in documents represented as mixtures of topics and topics represented as mixtures of words [28].

The first to utilize PLSA in information retrieval was Hofmann in 1999. In his paper, "Probabilistic Latent Semantic Analysis" (PLSA), Hofmann proposes and leverages the statistical techniques described above to build an IR algorithm called Probabilistic Latent Semantic Indexing. Hofmann applies PLSI to standard IR test sets, where it significantly outperforms both Boolean Retrieval and LSI [6]. PLSI has a couple of important drawbacks. First, the number of parameters grows linearly with the size of the corpus causing over fitting and reduced scalability. Second, PLSI does not sufficiently outline how to 'fold-in' documents, or create representations of documents outside of the training set [13].

Latent Dirichlet Allocation, proposed by Blei in 2003, builds upon the foundation set by LSI and PLSI. LDA is generally similar to PLSI, but with several statistical definitions that address problems in PLSI. For more on the statistical formulation and definition of LDA see [13]. Although, Blei developed LDA as a dimension reduction technique, LDA has been applied to IR. In "LDA-based Document Models for Ad-hoc Retrieval", Wei and Croft test LDA against Query-Likelihood (QL) retrieval and the Relevance Model (RM). An LDA based approach outperforms QL and matches RM on standard test sets [14].

V. Prior Work in IR with Philosophical Corpora

A. CARAT

In CARAT, De Pasquale and Meunier explore the use of perceptrons in what they call the 'categorization of small segments of text into a set of thematic categories' [9]. CARAT turns concept detection into a supervised learning problem, with labeled training data and test sets. They achieve surprising success with some thematic categories, reaching nearly 80% recall and 50% precision for the category 'knowledge'. However, their model is not universally successful, failing to reach much above 50% recall on the rest of their categories. Their mild success with a simple perceptron suggests that more complicated networks may be more successful at the categorization process.

However, supervised classification is not a robust solution to concept detection. Classification algorithms require large amounts of training data in order to be effective. Thus, labeled training data would need to be collected for every concept in a corpus. Collecting labeled data would be difficult, reducing the scalability of the system. Moreover, a supervised classification system would have no way of searching for a concept it has not been trained to find. Finally, since texts differ in structure and content according to author and subject, classifiers would not easily generalize to new corpora.

B. LDA Approaches

In 'Detecting Large Concept Extensions for Conceptual Anaylsis', Chartrand applies LDA to concept detection [10]. Chartrand notes important differences between concept detection and traditional IR. In traditional IR, all documents related to the concept expressed by the query are of interest to the user. In concept detection, the user is looking for all segments where the queried concept is present or all segments relevant to the conceptual analysis. He also highlights that latent semantic analysis and its descendants perform thematic rather than conceptual analysis. These algorithms express documents as mixtures of topics which *do not necessarily correspond with concepts*. Consider this example. If an author lists these words 'grape, apple, strawberry' in multiple documents, an algorithm like LDA will infer a topic that consist of the words 'grape, apple, and strawberry'. However, this topic does not represent a concept that the author expressed in their work, and simply indicates that several items are often listed together.

Chartrand uses topic models to infer the presence of concepts. The LDA based algorithm, searches for a concept's 'signifier' or 'concept-word' in the topic-word distribution of a topic and matches the topic with that concept based on the presence or absence of such signifying words within the topic. Documents highly associated with the identified topic are determined to be relevant to the query. The algorithms is evaluated on a corpus of law related documents. Labeled test data is obtained through crowd sourced tagging and expert judgement. While LDA achieves some success, it fails to score above 18% recall and 65% precision.

C. COFIH

To improve upon the results of the LDA approach, Chartrand and Meuiner developed a clustering based technique for concept detection: COFIH (Concept-Finding Heuristic) [11]. In COFIH, the corpus is converted into a co-occurence matrix. The queried concept is expressed as a document, consisting of signifiers or concept words, and added to the matrix. All documents containing at least one signifying term are extracted. This set is clustered and for each cluster a prototype or typical vector is built. The entire corpus is then checked for similar vectors to this prototype. The most similar vectors from all the clusters explored become the extension of the concept, or the set of segments relevant to the concept.

COFIH is evaluated on the concept detection task using a corpus of the collected papers of C.S. Pierce and compared against expert judgement. COFIH achieves 69% recall. An in-depth analysis of the results for the concept 'law', found the retrieved segments to be of a very high quality. However, COFIH returns more than 10 times the number of segments that term-matching yields. Moreover, the test set is tagged based only on the presence of a concept. Thus, the absence of a tag does not necessarily indicate the absence of a concept. As a result, precision is impossible to measure. According to its authors, COFIH shows promise but further validation is needed before significant conclusions can be drawn.

VI. Word Embedding

Word Embedding is a recent and important development in NLP. Word Embedding makes use of a vector space, representing each word as a unique vector, or "word-vector". These vectors are computed such that semantically similar words have vectors that are close together. LSI and VSM in part achieve this goal, generating vectors for each word in their vocabulary. In fact, a simple co-occurrence matrix can be seen as a form of word embedding albeit a low quality one. Modern word embedding techniques generate much higher quality word vectors. As such, they are a promising area of research in information retrieval and concept detection.

A. Word2Vec

Word2vec was proposed by Mikolov in 2013 and offers a novel, neural network based technique for learning embeddings. Word2vec's network is simple. It has a single input layer, a hidden 'embedding' layer, and a softmax output layer [3]. The 'embedding' layer is a set of hidden neurons of size d, a hyperparameter indicating the number of embedding dimensions. The dense connections between the input layer and the hidden layer form a matrix of weights, of size $n \times d$, where n is the size of the vocabulary and d is the number of

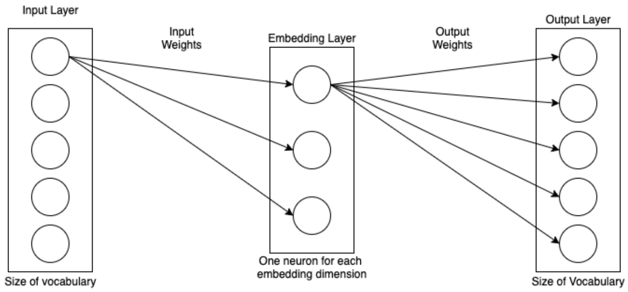

Fig. 1: Word2vec's architecture

dimensions in the embedding space. Each row of this matrix is a word embedding, or word vector, for a word in the corpus. During training, these weights are altered, computing high quality embeddings for each word.

The most commonly used training scheme for word2vec is skip-gram. In this schema, a sliding window is passed over the corpus. The center word, in the middle of the window, is fed into the model. The model then tries to predict the other words present in the window. This process alters the embedding of each word in the hidden layer. After successfully training the model, semantically similar words are represented as vectors that are close to each other in the semantic space [3].

These vectors have powerful properties. Consider the following example with the words king, queen, prince, princess, pear, and apple, whose embeddings are shown in two dimensions in Figure 2. Word2vec is able to compute vectors that express both semantic similarities and semantic relationships between words. In the embedding space in Figure 2, the words are grouped into two clusters: [king, queen, prince, princess] and [pear, apple]. The clusters are semantically similar, with one containing royal titles and another containing fruits. However, a closer look at the royalty cluster shows that the distance between king and queen is exactly the same as the distance between prince and princess. In fact, subtracting king's vector from queen's, yields a vector that, when added to prince's, results in a vector similar to the one for princess. The embeddings word2vec generates capture analogies between words in terms of the distance between vectors. In Mikolv's work, word2vec was able to capture complex semantic analogies much more effectively than previous embedding approaches [3]

One explanation for the success of word2vec lies in the distributional hypothesis. In skip-gram training, word2vec predicts words based on their context. Thus, word2vec learns the syntactic structure of text and the internal weights in the embedding layer are trained accordingly. However, as outlined by Mikolov, in addition to syntactic structure, the model learns an accurate representation of the semantic structure of text [3]. The distributional hypothesis asserts that syntactic structures reflect semantic structures in natural language. This explains why word2vec is able to learn high quality semantic word embeddings despite training on syntactic data.

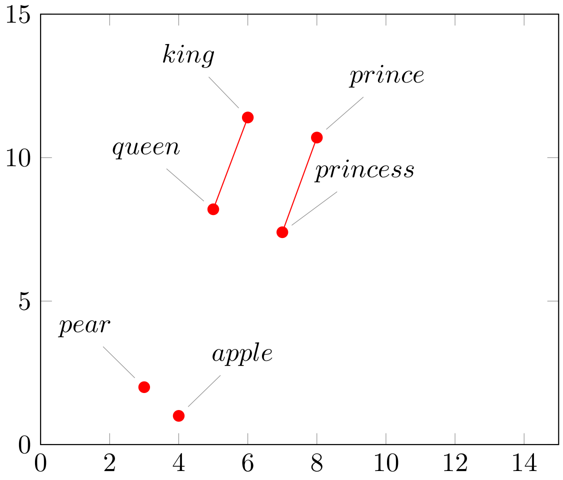

Fig. 2: Example of word2vec embeddings in 2D space

B. Application of Word Embedding to IR and Concept Detection

Word embedding techniques are able to compute high quality word vectors that capture complex semantic information from natural language. These word vectors provide a promising solution to the issues of synonymy and polysemy in Information Retrieval. IR techniques leveraging word embeddings would have a robust understanding of semantics and be better equipped to deal with both issues. Additionally, word embedding provides a promising tool for further research in concept detection, which deals with semantically complex concepts.

VII. Average Cosine Similarity (ACS): A Simple Word Embedding Driven Algorithm for Concept Detection

This section proposes Average Cosine Similarity (ACS) a simple, word embedding driven retrieval algorithm for concept detection. The user provides a conceptual query comprising a single key word, e.g. [FREEDOM], [EVIL], [JUSTICE], corresponding to a specific concept. Document relevance to the concept is calculated as the average cosine similarity between the component words of the document and the query's key word. Given a queried key word represented by the word vector q and given a document d consisting of words represented by the word vectors $[w_1, w_2 ... w_n]$, the relevance of the document to the query is calculated as:

$$\frac{1}{n}\sum_{i=1}^{n}\frac{q \cdot w_n}{||q|| ||w_n||} \tag{5}$$

Each segment in the corpus is scored and the top segments are returned to the user. In practice, small segments are ignored as their length makes the ACS calculation volatile. Our version of ACS labels all segments under 30 words in length 'Irrelevant'.

ACS can be conceptualized as a nuanced key word search, leveraging not only the key word, but also the most similar words to the key word in the word2vec model. Because these words co-occur frequently with the key word in the corpus, it is likely that they represent either characteristics or extensions of the concept. Therefore, the presence of these words could indicate the presence of the concept.

If this assumption is correct, ACS has the potential to improve upon key word search in two areas. Key word search struggles with segments that are relevant to the concept but do not contain the key word. Segments like these will contain many words related to the concept. If these words have high similarity scores, ACS will give the segment a high score even though the key word is absent. Key word search also struggles with segments that are not relevant to the concept but contain the key word. It is likely that these segments will contain many words unrelated to the concept. If these unrelated words have low key word similarity scores, then ACS will assign the segment a low score despite the presence of the key word.

VIII. Word Mover's Distance for Concept Detection

Although ACS leverages words similar to the key word, there is no guarantee that these word represent the characteristics or extensions of a concept. Thus, algorithms that take groups of words as queries provide a promising evolution from the one word approach of ACS and key word.

Word Movers Distance, as proposed by Kusner, Sun, Kolkin, and Weinberger in 2015, is a word embedding driven algorithm for calculating the distance between documents [32]. Word Mover's Distance is a special case of the Earth Mover's Distance or Wasserstein Metric, which is used to measure the similarity of two distributions as the work required to 'cover' one distribution with the other. For example, imagine five piles of dirt and five holes. The work required to take the dirt from the piles and fill all the holes is the weight of the dirt times the shortest possible distance it has to be carried.

Word Movers Distance applies this algorithm using word distributions and word embedding to estimate the work required to turn one segment into another. WMD represents segments as distributions of words using the normalized bag-of-words method (nBOW). The distance between words is calculated as the euclidean distance, $||w1 - w2||$, between the embeddings of the two words. EMD is then used to calculate the distance between documents [32]. Figure 3 visually illustrates this calculation. WMD was shown to have high performance on document categorization tasks, outperforming state of the art techniques [32].

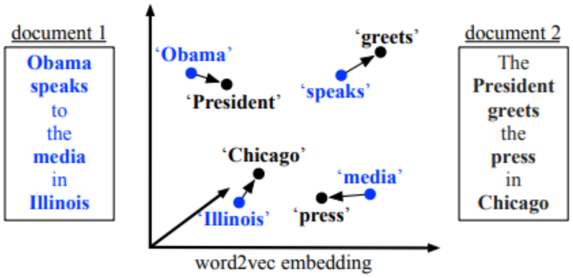

Fig. 3: WMD illustration in "From Word Embeddings to Document Distances"

Our WMD retrieval algorithm for concept detection ranks segments by their similarity to a canonical definition of the concept. The user provides a query consisting of a segment from the corpus that defines the concept of interest. Each segment in the corpus, except for the query segment, is given a score based on its WMD similarity to the query segment. Stop words are removed from both the query and the segments in the corpus before calculating similarity. The top segments are returned to the user.

Unlike ACS, WMD explicitly requires the user to provide a set of words that represent characteristics and extensions of the concept. The query is guaranteed to be a segment that expresses the concept. Thus, segments that are similar to the query, likely define or relate to the concept. The definition segment also contains the fingerprint of the author's definition style - words like 'defines' and 'relates' that structure the author's explanations. Other documents that express definition will contain this fingerprint and have a high similarity to the definition segment.

IX. A Preliminary Word2Vec Model trained on the Digital Ricoeur Corpus (DR-1)

Both WMD and ACS rely on high quality embeddings. Thus, it is important to not only train a word embedding model on the Digital Ricoeur corpus but also to asses the quality of its embeddings. In the fall semester, the Digital Ricoeur corpus was pre-processed and a preliminary word2vec model, DR-1, was trained.

A. *Pre-processing for Word2Vec Training Data*

The Digital Ricoeur Corpus was tokenized into sentences. These sentences were further tokenized into lists of words. The corpus was not lemmatized or stemmed. We were able to keep preprocesing simple because of the power of word2vec's embeddings. There is no need to stem text when your embeddings capture semantic relationships between words.

B. *Training*

DR-1, our preliminary word2vec word embedding model, was trained on the Digital Ricoeur corpus using the skip-gram strategy. The model was trained on all texts in the corpus for 200 epochs with an embedding size of 300.

C. *Preliminary Model Evaluation*

The model was evaluated using the semantic quality of the word similarities in its embedding space. For each word in a subset of the important words and concepts in the Digital Ricoeur test set, the top 20 most similar words in the embedding space were computed. The order of the terms in these lists and the cosine similarity of each term was analyzed to determine if each reflected the semantic relationships in the corpus.

An analysis of word similarities in the trained model yielded two findings. The order of terms was strong. In general, the words with the highest cosine similarities had high semantic similarity. For example, the top scoring words for 'deiu' or god were:

1) divin [divine], 0.41136568784713745
2) peuple [people], 0.39945873618125916
3) yhwh [likely yahweh - lord], 0.39937764406204224
4) yahvé [lord], 0.3770883083343506
5) seigneur [lord], 0.37235385179519653
6) divine [divine], 0.37207528948783875
7) toi [you], 0.36198845505714417
8) dieux [gods], 0.3611795902252197
9) père [Father], 0.3604251742362976
10) péché [sin], 0.35684192180633545

While the order of terms was promising, the cosine similarities scores did not accurately reflect the semantic relationships in the text. For instance, divin [divine], a synonym for god, has a score of 0.41 while peuple [people], a highly related but distinct term has a score of 0.39. In addition, similarity scores for even the top words never reach higher than 0.5-0.6 despite their semantic similarity.

D. *Exploring the Behavior of ACS using the Concept Myth in "The Symbolism of Evil"*

Despite the drawbacks of DR-1's embeddings, the model made it possible to implement and run ACS and WMD on the Digital Ricoeur corpus. We began with ACS, as it is the simpler of the two algorithms. Rather, than fully analyze the results of this search for precision and recall, we chose to explore the results of ACS through the lens of three segment characteristics: rank, length, and key word presence. We chose these variables to explore several hypotheses about the ACS algorithm. First, an average is effected by the number of values in its set, thus we wanted to see how document length affected a segments rank. Second, since the user may only query a single word using ACS, we wanted to determine if the presence of that word had a strong effect on a segment's rank.

ACS was run on Ricoeur's book "The Symbolism of Evil" using the query [Mythe] and all paragraphs were scored and ranked by ACS. Figure 4 graphs rank vs length for all paragraphs in the work. Red dots indicate the segment contains the key word ‘Mythe’.

The results show a significant relationship between rank and presence of key word. The red segments carrying the key word are clustered toward the left hand side of the graph at high ranks. This relationship could relate to issues with the similarity scores in the DR-1 model. DR-1’s similarities scores for even the most similar words to the key word never reach higher than 0.5-0.6 while the similarity score from the key word to itself is always 1. Thus, the key word has a disproportionately strong effect on the ACS score of a segment.

The results also show a possible relationship between the rank of a segment and the length of a segment. Towards the left hand side of the graph there is a slight curve, indicating a gradual increase in segment length. A relationship between rank and length could be explained by the characteristics of the ACS score computation. Consider two documents, D_1, D_2, where $len(D_1) > len(D_2)$. In the case that they are both relevant to the query, the preliminary algorithm must assign them similar high scores. In order for this to occur, the proportion of terms semantically related to the query to terms not related must be constant in both documents. However, as document size grows, this is often not the case. In long documents, stop words and connectors — for, and, the, so — make up greater and greater portions of the text. Thus, in bigger documents, words semantically related to the concept have less and less of an impact on the average cosine distance to the concept word.

X. Improving the Preliminary Word Embedding Model with Pre-training

As explained in Section IX-C, the preliminary word embedding model did not sufficiently capture the semantic relationships between words in the Ricoeur corpus. It is common in domain specific applications, where training corpora is limited in size, for word embedding models to struggle. Mikolov notes that corpora size heavily impacts embedding quality, “training on twice as much data using one epoch gives comparable or better results than iterating over the same data for three epochs” [3]. At just over 3M tokens, the Digital Ricoeur corpus is much smaller than traditional word2vec training sets, which number in the billions of tokens [3].

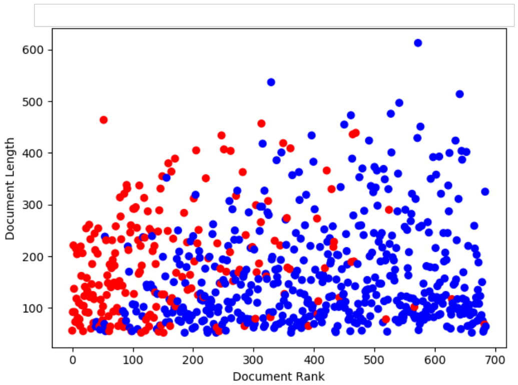

Fig. 4: Rank vs Length and Key Word in the Results for Myth

We examined several methods for improving our model, despite the limits of the Digital Ricoeur corpus. The corpus could be augmented with works from similar philosophers, related scholars, or general purpose text documents. However, extending the corpus greatly increases the computational resources required to train the model. In addition, to reach a corpus size on the order of billions of tokens, the majority of the corpus would have to filled with out-of-domain texts. Thus, the resulting model would likely be general purpose. Training a general purpose model is redundant as many already exist in open-source formats.

Pre-training provides a stronger alternative. A pre-trained word2vec model is initialized with embeddings trained on a large, generalized corpus and then trained using domain specific texts. The strength of this approach is that it allows the model to leverage general semantic knowledge while preserving the domain specificity of the its embeddings by limiting training to texts in the domain. This procedure produces vectors that have both a global and a domain specific representation of natural language.

A new Word2Vec model, DR-2, was developed using pre-training. DR-2 was initialized with pre-trained embeddings from NLPL’s Model 43 trained on the French CoNLL17 corpus and then trained on the Digital Ricoeur corpus for 200 epochs using the skip-gram strategy [31]. DR-2 has an embedding dimension of 100 in order to match the embedding dimension for the NLPL model. An exploration of the important terms in DR-2 showed a significant improvement in the quality of the model.

Both the model's ordering of terms and its similarities scores were analyzed. Table I shows the Top 10 most similar terms for 'symbole' in both DR-1 and DR-2.

Word	DR-2 (pre-trained)	DR-1 (not pre-trained)
1	**'symbolisme', 0.741**	**'mythe', 0.515**
2	**'mythe', 0.717**	**'symbolisme', 0.493**
3	'schème', 0.563	'sens', 0.406
4	'langage', 0.545	'langage', 0.398
5	'péché', 0.537	**'serpent', 0.363**
6	'sacré', 0.535	'mal', 0.343
7	'mal', 0.532	'péché', 0.340
8	'sens', 0.525	'schème', 0.336
9	'mouvement', 0.524	'symbolique', 0.330
10	**'serpent', 0.5239921808242798**	'mot', 0.329

TABLE I: The Top 10 Most Similar Words to Symbole in the Pre-trained Model and Preliminary Model

The pre-trained model's similarity scores are much higher. The scores of the top 15 most similar words to 'symbol' have an average percent change of 53% from DR-1 to DR-2. These higher scores lessen the bias toward the key word evident in the preliminary model. Moreover, the similarity scores of DR-2 better reflect the semantic relationships in the text. The word 'symbolism' is closer than the word 'meaning' to 'symbole'. The percent difference between the similarity scores of 'symbolism' and 'meaning' in the pre-trained model (34%) is 10% greater than in the preliminary model (25%).

The order of terms has also improved. While 'symbolism' and 'myth' are both closely related to 'symbol', 'symbolism' is a closer semantic match. Words like 'serpent', which refer to specific symbols are not as closely related to 'symbol' as words like 'language' (langage) and 'meaning' (sens). DR-2 captures both of these relationships, ranking 'symbol' higher than 'myth' and 'language/meaning' higher then 'serpent'. DR-1 flips both these relationships.

A similar analysis completed for the top 10 most similar words for homme (man), mythe (myth), justice, and morale (moral) found similar improvements in DR-2.

XI. Quantifying Bias in ACS using DR-2

With a higher quality word embedding model, DR-2, we decided to re-evaluate ACS to determine if the biases discussed in Section IX-D were evident in the results of ACS using DR-2. While Section IX-D relies on a graph of rank vs length, this section evaluates biases using correlation coefficents.

A. *Exploring Bias with Linear Correlation Metrics*

To explore these hypotheses, ACS was run using the new, pre-trained embeddings on all segments from both "The Symbolism of Evil" (SM) and "One's self as Another" (SA). Each segment was ranked to several uni-gram concepts - mythe, symbole, and homme for SM; justice and morale for SA. All segments were ranked according to their ACS relevance score. A Pearson Correlation was used to determine if there was a linear relationship between a segment's length and its rank. A Point Biserial Correlation, a Pearson Correlation between a binary, categorical variable and a continuous variable, was used to determine if there was a linear relationship between rank and presence of keyword. In each case, the null hypothesis assumed no correlation between the variables.

B. *Results*

Concept	Rank vs Length	Rank vs Concept Word
Mythe	r=0.084 p=0.017	r=-0.499 p<0.001
Symbole	r=0.092 p=0.01	r=-0.339 p<0.001
Homme	r=-0.028 p=0.421	r=-0.382 p<0.001
Morale	r=0.008 p=0.796	r=-0.363 p<0.001
Justice	r=-0.035 p=0.309	r=-0.434 p<0.001

TABLE II: R and P values for Rank vs Length and Concept Word for all Concepts

The results show no statistically significant correlation between rank and length in the concepts homme, justice, and morale ($p > 0.05$). The results for mythe and symbole show very weak ($r = 0.084$, $r = 0.092$) yet statistically significant linear correlations between rank and length ($p < 0.05$). However, the r value for these concepts is low enough that such a correlation can be considered negligible. Thus, rank and length have no meaningful correlation in the results of ACS across the 5 concepts, indicating ACS exhibits no substantial bias towards segment length.

The results for rank vs keyword display a statistically significant and moderate negative linear correlation between the variables ($p < 0.001$, $-0.363 > r > -0.499$). The negative r indicates the presence of the key word correlates to a higher rank.

Both of these relationships can be seen in the results for justice, displayed in Figure 5. Segments of all lengths are distributed equally in the rankings. Red dots, indicating segments with the key word, are clustered towards the left hand side of the graph at higher ranks. Thus, while ACS shows no meaningful bias towards segments length, it exhibits a substantial bias toward key word.

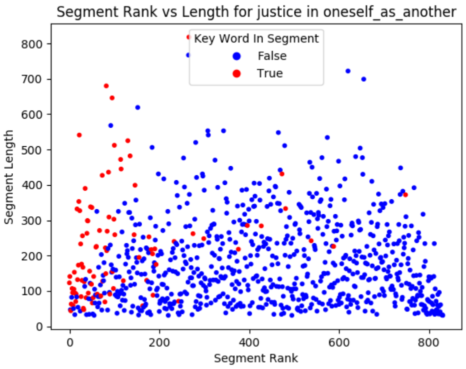

Fig. 5: Segment Rank vs Length for Justice in SA. Red dots indicate segments with the key word.

C. Modifying ACS: Mitigating Bias Towards Key Word

Three more experiments were conducted using modified versions of ACS designed to minimize key word bias. Trial one (“stopwords”) removes stop words, decreasing the number of noise words in each segment. Trial two (“projection”) takes the difference between the top most similar words to the key word, adds that difference to the similarity of the most similar word, and sets that new value as the similarity of the key word to itself. This step decreases the influence of the key word in the calculation of the average. Trial three (“removal”) removes the key word from the ACS calculation. Both trial two and trial three also remove stop words. Each of the modified ACS algorithms was run on all concepts from both SM and SA. Pearson Correlations and Point Biserial Correlations were used to determine if there was a linear relationship between rank and length and rank and key word. In each case, the null hypothesis assumed no correlation between the variables

D. Modified ACS Results

Results for rank vs length for all three modified ACS trials indicate no meaningful correlation between rank and length ($p > 0.05$, $r < 0.1$). Results for rank vs key word presence for all three modification trials (1, 2, and 3) as well as the baseline trial (0) are displayed in Figure III. The p-values for all trials were < 0.05 and therefore statistically relevant. All trials exhibited r values of -0.2 or less, indicating low to moderate correlations. The average percent change between trials is shown in Figure IV. Stop word removal increased the magnitude of the correlations by an average of 10%, projection of the key word had little to no effect on the magnitude of the correlations, and removal of the key word decreased the magnitude of the correlations by an average of 29%. Thus, trial 1 amplified key word bias, trial 2 had no affect on key word bias, and trial 3 significantly decreased key word bias.

The significant decrease in the magnitude of r in trial 3, shows that instances of the key word have a large part in causing the correlation between rank and key word. However, removal of the key word does not completely eliminate this correlation - trail 3 still exhibits a weak to low correlation ($-0.212 > r > -0.372$). Words that occur in similar contexts to the key word, also have a strong affect on the ACS score. Segments where the key word appears contain more of these kinds of words and therefore receive higher ACS scores. Thus, even when the key word is ignored, segments that contain the key word receive high scores.

Concept	Trial 0	Trial 1	Trial 2	Trial 3
Modification	None	stopwords	projection	removal
Mythe	r=-0.499	r=-0.507	r=-0.472	r=-0.270
Symbole	r=-0.339	r=-0.368	r=-0.334	r=-0.213
Homme	r=-0.382	r=-0.450	r=-0.374	r=-0.212
Morale	r=-0.363	r=-0.443	r=-0.416	r=-0.334
Justice	r=-0.434	r=-0.452	r=-0.426	r=-0.372

TABLE III: R values for Rank vs Key Word for all Concept and Trials. $p < 0.05$ for all r.

	Trial 1	Trial 2	Trial 3
Avg Percent Change	10.8%	$\approx 0\%$	-0.29%

TABLE IV: Average percent change in R between the modified ACS trials

XII. Exploring the Effect of Corpus Quality on Bias in ACS

In Section XI and XI-C, we investigate algorithmic causes of bias in ACS. However, the behavior of ACS could be explained by another variable: corpus quality. The Digital Ricoeur corpus is composed of digitized pdfs converted to plain text using OCR. As a result, the corpus contains OCR errors - misidentified characters that cause misspellings and omissions. To assess the impact of corpus quality on the biases in ACS, an alternative, clean corpus of philosophical texts was obtained from the Project Gutenberg philosophy bookshelf [33]. The corpus contains 94 texts written by influential philosophers from ancient and modern history - eg. Plato, Aristotle, Locke, Nietzsche - in english. The corpus contains 11.9M tokens and has a vocabulary size of 119,557.

A Word2Vec model was initialized using pretrained embeddings from NLPL's english model 40 trained on two iterations of the English CoNLL17 corpus [31]. The model was then trained on the Gutenberg corpus for 10 iterations. ACS was used to rank paragraphs from Aristotle's "Nicomachean Ethics" according to three concepts: good, moral, and virtue. Just as with Digital Ricoeur's corpus, correlation coefficients were calculated for both segment rank vs segment length and segment rank vs presence of key word for each concept. These results are displayed in Table V. In each case, the null hypothesis assumed no correlation between the variables.

Concept	Rank vs Length	Rank vs Key Word
Good	r=0.038 p=0.216	r=-0.500 p<0.001
Moral	r=-0.068 p=0.027	r=-0.381 p<0.001
Virtue	r=0.025, p=0.412	r=-0.262 p<0.001

TABLE V: R and P values for Rank vs Length and Key Word in the Gutenberg Corpus

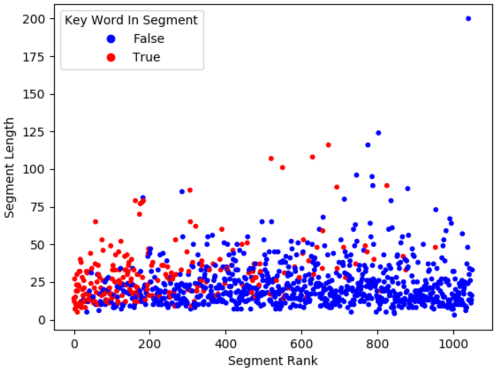

Fig. 6: Segment Rank vs Length for Good in Aristotle's Nicomachean Ethics. Red dots indicate segments with the key word.

The statistical analysis shows that the results of ACS on the Project Gutenberg corpus exhibit no meaningful correlation between rank and length. The r values from Virtue and Good have p-values greater than 0.05 and thus are not statistically significant. The r value for Moral is statistically significant but it does not represent a meaningful correlation ($p < 0.05$, $|r| < 0.1$). Just as in the Ricoeur experiment, the r values for rank and key word from the Project Gutenberg experiment indicate a moderate to low, statistically significant correlation ($p < 0.05$, $0.262 < |r| < 0.5$). Thus, corpus quality seems to have no effect on the biases in ACS.

XIII. Evaluating the Effectiveness of ACS and WMD Retrieval using the Digital Ricoeur Test Set

The performance of ACS and WMD retrieval on the task of concept detection was evaluated against Key Word search (KW) using the Digital Ricoeur test set described in Section III. Four versions of ACS, two versions of WMD, and Key Word search were used to rank all segments in the test set.

Key Word (KW): In order to have a uniform analysis, a ranking version of key word search was used. Segments were given a score based on the number of occurrences of the key word in the segment. Segments were then ranked using this score.

Average Cosine Similarity (ACS): Each of the four modifications of ACS from section XI-C were used to rank the segments.

- **Trial 0** had no modifications.
- **Trial 1** excluded stopwords from the ACS calculation
- **Trial 2** excluded stopwords and set the similarity of the key word to itself to the difference between the top two most similar words to the key word plus the similarity of the most similar word to the key word. This was informally called "projection".
- **Trial 3** excluded stop words and removed the key word from the ACS calculation.

Word Mover's Distance Retrieval (WMD): WMD was performed with two different sets of definitions. WMD-1 was run using segments experts tagged as 'defines'. WMD-2 was run with canonical, hand-picked, definitions. In both WMD trials, the pre-trained embedding model was used. All definition segments that were part of the test set were removed before calculating recall and precision. Stop word lists were obtained from Spacy's 'fr_core_news_sm' model [34].

The differences in the scoring mechanisms of ACS, WMD, and KW complicated our analysis. Typically, binary classification algorithms that produce a score can be evaluated using a precision-recall curve, in which precision and recall are calculated at every possible threshold. This strategy works well for WMD and ACS, which produce continuous scores. However, KW produces a discrete score representing the number of key words in the document. Thus, WMD, ACS and KW cannot be compared using a precision-recall curve.

We chose to evaluate the retrieval strategies using a Top N approach. All six algorithms were used to score all segments in the Digital Ricoeur test set to the corresponding concepts. Segments from SM were ranked for the concepts mythe, symbole, and homme. Segments from SA were ranked for the concepts morale and justice. For each retrieval strategy, the top 5, 10, 15, and 20 segments were returned as ‘relevant’ and precision/recall was calculated for each set (top 5, top 10). Note, however, that precision is proportional to recall because the number of positives (Top N) is kept constant. This approach simulates and evaluates the use of each strategy as a search rank algorithm. Precision at the the top n threshold represents the percentage of relevant documents among the n documents returned by the algorithm.

After completing the experiment, we chose to exclude the results from SA because of several issues with the test set. First, SA was ranked by 3 rather than 4 judges. Thus, the standard for relevance is not constant between both books. Segments in SM require 3 ‘Relevant’ tags to be considered ‘Relevant’ while segments from SA only require 2. Additionally, ‘morale’ and ‘justice’ are ill defined in Ricoeur scholarship. Thus, there is less of a scholarly consensus as to the intention and extension of these concepts as well as a greater degree of disagreement among judges about the definition of the concepts.

XIV. Results

A. Top 5

Results at the top 5 threshold are displayed in Table VI. Results are color coded based on their relationship to keyword - Green: beats KW, Blue: ties KW, and Red: loses to KW. WMD-2 outperforms KW in a majority of concepts. WMD-1 is outperformed by KW in a majority of concepts. Trial-1 and Trial-0 beat KW in a majority of concepts. Trial-1 performs slightly better than Trial-0. Trials 2 and 3 tie and lose to KW respectively. Table VII compares the precision values of the best strategies, Trial-1 and WMD-2. WMD-2 outperforms Trial-1 in a majority of concepts.

	Mythe	Symbole	Homme
trial-0	0.8	0.4	0.4
trial-1	0.8	0.6	0.2
trial-2	0.8	0.3	0.2
trial-3	0.5	0	0.2
KW	0.6	0.6	0.2
WMD-1	0.6	0.4	0
WMD-2	1	0.8	0.2

TABLE VI: Precision among Top 5 Segments

	Mythe	Symbole	Homme
Trial 1	0.8	0.6	0.2
WMD 2	1.0	0.8	0.2

TABLE VII: Precision among Top 5 for Top 2 Strategies

B. Top 10

Results at the top 10 threshold are displayed in Table VIII. Results are color coded based on their relationship to keyword - Green: beats KW, Blue: ties KW, and Red: loses to KW. WMD-2 outperforms KW in a majority of concepts. WMD-1 is outperformed by KW in all concepts. Trial-1 ties KW in two concepts and loses in one concept. Trials 0, 2, and 3, either tie or lose to KW in all concepts. Table VII compares the precision values of the best strategies, Trial-1 and WMD-2. WMD-2 outperforms Trial-1 in a majority of concepts.

	Mythe	Symbole	Homme
trial-0	0.8	0.2	0.2
trial-1	0.8	0.3	0.2
trial-2	0.7	0.3	0.2
trial-3	0.3	0.1	0.2
KW	0.8	0.5	0.2
WMD-1	0.7	0.3	0
WMD-2	0.8	0.7	0.3

TABLE VIII: Precision among Top 10 Segments

	Mythe	Symbole	Homme
Trial-1	0.8	0.3	0.2
WMD-2	0.8	0.7	0.3

TABLE IX: Precision among Top 10 Segments for Top 2 Strategies

C. Beyond Top 10

Results at Top 15 threshold and Top 20 threshold are displayed in tables X and XI. At higher thresholds both ACS and WMD are both outperformed by KW in a majority of concepts.

	Mythe	Symbole	Homme
trial-0	0.87	0.26	0.2
trial-1	0.75	0.25	0.26
trial-2	0.75	0.25	0.2
trial-3	0.37	0.12	0.2
KW	0.75	0.66	0.33
WMD-1	0.73	0.37	0.06
WMD-2	0.866	0.6	0.26

TABLE X: Precision among Top 15 Segments

	Mythe	Symbole	Homme
trial-0	0.71	0.2	0.15
trial-1	0.66	0.28	0.25
trial-2	0.61	0.23	0.19
trial-3	0.33	0.19	0.14
KW	0.8	0.7	0.25
WMD-1	0.6	0.33	0.09
WMD-2	0.85	0.65	0.2

TABLE XI: Precision among Top 20 Segments

XV. Discussion

A. Removal of Stopwords Causes Significant Improvement in ACS

ACS with stopword exclusion, Trial-1, is the highest performing ACS trial. Among the top 5 segments, Trial-1 has a higher precision values than Trial-2 in 'Symbole' and the same value in 'Mythe' and 'Homme'. Trial-1 has a higher precision value than Trial-0 in 'Symbole' and 'Mythe' and a lower value in 'Homme'. Trial-1 has a higher precision value than Trial-3 in 'Mythe' and 'Symbole' and the same value in 'Homme'. Among the top 10 segments, Trial-1 has higher precision than all other ACS trials for the concept 'Mythe'. Trial-1 has higher precision than Trial-0 and Trial-3 for the concept 'Symbole'. Trial-1 has the same precision as Trial-2 for the concept 'Symbole'. Trial-1 has the same precision as all other ACS trials for 'Homme'. Thus, tt both the top 5 and top 10 thresholds, ACS Trial-1 has the best performance of all ACS trials on the concept detection task in the Digital Ricoeur test set.

B. WMD Outperforms ACS

WMD-2 significantly outperforms WMD-1, producing higher precision values for every concept, at both top 5 and top 10. WMD-2 also significantly outperforms ACS Trail-1, the best ACS trial. WMD-2 has higher precision values for 2 out of three concepts at top 5 and top 10. Therefore, WMD-2 has the best performance of all word embedding driven algorithms on the concept detection task in the Digital Ricoeur test set.

C. WMD Outperforms KW on the Concept Detection Task

Not only does WMD-2 outperform ACS, it also produces better results than KW at both the top 5 and top 10 thresholds. At top 5, WMD-2 has higher precision in 'Mythe' and 'Symbole' and the same precision in 'Homme'. At top 10, WMD-2 has higher precision in 'Symbole' and 'Homme' and the same precision in 'Mythe'. Therefore, WMD-2 matches and exceeds the performance of KW at both thresholds.

WMD also performs significantly better compared to KW at top 5 and top 10 than it does at top 15 and top 20. At both 15 and 20, WMD-2 has lower precision than KW in a majority of concepts. However, this finding is less relevant because of the constraints of concept detection. As mentioned in Section I-C, returning 15 to 20 segments, would be impossible because of fair-use copyright law, which restricts the percentage of the corpus that can be returned from a search. Moreover, a high volume of segments runs the risk of overloading the user. WMD-2's better performance at lower top n thresholds actually means it is well suited for the task of concept detection.

WMD-2's performance on the Digital Ricoeur test set indicates that WMD retrieval is well suited to the task of concept detection and capable of outperforming common techniques like key word search. Thus, WMD retrieval is a promising algorithm that could improve the performance of concept detection tools in theoretical corpora. This finding is applicable to databases like Digital Ricoeur and JSTOR which rely on robust search tools to help their users conduct research.

XVI. Future Work

A. Expanding the Test Set

Although WMD produces promising results on the Digital Ricoeur test set, the scope of this study is limited to one book and 3 concepts. To fully assess the performance of WMD retrieval on concept detection, a larger test set is necessary. However, obtaining test data is difficult. Tagging complex, theoretical corpora requires time and resources. As evidenced by the issues with the test set extracted from One's Self as Another, it can be difficult to find willing experts and to get those experts to agree on conceptual definitions. Crowd sourcing provides a solution to this problem. While crowd sourcing dilutes the knowledge base of the judges, it drastically increases the size of the test set and allows the judgements to come directly from a relevant set of users. A crowd sourcing tool for the Digital Ricoeur website could be developed. The tool would let users of the database tag segments according to concepts. Given enough time, such a tool could drastically increase the size of the Digital Ricoeur test set.

B. New Uses of the Digital Ricoeur Test Set

In addition to providing ground truth values for concept detection tasks, the Digital Ricoeur test set provides an exciting opportunity to explore the linguistic properties of abstract theoretical concepts. The segments deemed relevant to a concept could be analyzed to uncover patterns in syntactic and semantic information. Analyzing these sets could also reveal the characteristics and extensions of the concept. The test set could also be used to assess the performance of document clustering algorithms, like kNN, on a theoretical corpus.

C. Evaluating and Optimizing the Word2vec Model

Although pre-training significantly improved the quality of the word2vec model, its embeddings are far from perfect. Before attempting to improve the quality of the embeddings automated tests of embedding quality should be developed. One solution is an analogy test, which tests the models ability to capture relationships between words. Although there are generalized, open source analogy tests, the model is trained on a specialized corpus and therefore a Digital Ricoeur analogy test would be most useful. Development of such a test would likely require a team of scholarly experts.

With an effective performance test for the word2vec model, different parameters can be explored in order to optimize the model's quality. In this study two hyper parameters of the word embedding model are not exhaustively studied: the number of training epochs and the embedding size. A thorough examination of these variables could lead to an improvement in the mode. In addition, word2vec is not the only algorithm for generating embeddings. GloVe, a powerful word embedding algorithm developed by researchers at Stanford, has been shown to produce better embeddings than word2vec [4].

D. Transforming Concept Detection with New NLP Technologies

In recent years, several new technologies have rocked the NLP world. One such technology, transformers, provides an exciting avenue of further research within concept detection. Transformers translate sequences of text from one encoding into another. Transformers have been applied throughout the NLP world with great success. Transformers also make up new state-of-the-art pre-trained models like BERT and ELMO.

E. WMD Beyond Concept Detection

The effectiveness of WMD on the concept detection task indicates that it may have wide ranging applicability among IR problems. WMD retrieval could be explored in new domains - medical texts, scientific abstracts, news articles, or web documents. WMD's ability to accurately estimate the similarity of text segments opens up promising use cases within the Digital Ricoeur Project. During the course of research, a scholar often finds themselves looking for more information related to the text they are currently exploring. WMD retrieval could act as a 'Related Material' search, in which the researcher provides the system with a text segment of interest and the system returns similar segments. If proved effective, WMD could help researchers quickly navigate the Digital Ricoeur corpus.

XVII. Acknowledgements

This study would not have been possible without the help of faculty advisors: Fernando Nascimento, Stephen Majercik, and Stacy Doore. We would also like to recognize several key libraries and resources we used in our research: SpaCy, Gensim, and NLPL.

References

[1] J. Firth, "A Synopsis of Linguistic Theory 1930-1955," *Studies in Linguistic Analysis*, vol. Special Volume of the Philological Society, pp. 1–32, 1957.

[2] Salton and McGill, *Introduction to Modern Information Retrieval*. New York, NY, United States: McGraw-Hill, 1987.

[3] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, *Efficient Estimation of Word Representations in Vector Space*. 2013.

[4] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global Vectors for Word Representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543, [Online]. Available: http://www.aclweb.org/anthology/D14-1162.

[5] S. Deerwester, S. Dumanis, G. Furnas, T. Landauer, and R. Harshman, “Indexing by Latent Semantic Analysis,” *Journal of the American Society for Information Science*, vol. 41, no. 6, pp. 391–407, 1990.

[6] T. Hofmann, “Probabilistic Latent Semantic Analysis,” in *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence*, San Fransisco, CA, 1999, pp. 289–296.

[7] S. P. Crain, K. Zhou, S. H. Yang, and H. Zha, “Dimensionality Reduction and Topic Modeling: From Latent Semantic Indexing to Latent Dirichlet Allocation and Beyond,” in Mining Text Data, Boston, MA: Springer.

[8] D. Ganguly, D. Roy, M. Mitra, and G. Jones, “Word Embedding based Generalized Language Model for Information Retrieval,” in *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2015, pp. 795–798, [Online]. Available: https://doi.org/10.1145/2766462.2767780.

[9] J.-F. De Pasquale and J.-G. Meunier, “Categorisation Techniques in Computer-Assisted Reading and Analysis of Texts (CARAT) in the Humanities,” *Computers and the Humanities*, vol. 37, no. 1, pp. 111–118, 2003.

[10] L. Chartrand, J. C. K. Cheung, and M. Bouguessa, “Detecting Large Concept Extensions for Conceptual Analysis,” in *Machine Learning and Data Mining in Pattern Recognition*, 2017, pp. 78–90.

[11] L. Chartrand et al., “CoFiH: A heuristic for concept discovery in computer assisted conceptual analysis,” in *Proceedings of 13ème Journées internationales d’Analyse statistique des Données Textuelles*, JADT, 2016.

[12] A. Berger and J. Lafferty, “Information Retrieval as Statistical Translation,” in *Proceedings of the 22nd annual international ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999, pp. 222–229, [Online]. Available: https://doi.org/10.1145/312624.312681.

[13] D. Blei, A. Ng, and M. Jordan, “Latent Dirichlet Allocation,” *The Journal of Machine Learning Research*, vol. 3, 2003.

[14] X. Wei and W. Croft, “LDA-based document models for Ad-hoc retrieval,” in *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, United States, Jan. 2006, pp. 178–185, doi: 10.1145/1148170.

[15] X. Yi and J. Allan, “Evaluating Topic Models for Information Retrieval,” in *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, New York, NY, United States, Oct. 2008, pp. 1431–1432, doi: 10.1145/1458082.

[16] C. Zhai, “Statistical Language Models for Information Retrieval,” *Synthesis Lectures on Human Language Technologies*, vol. 1, no. 1, pp. 1–141, Jan. 2008, doi: 10.2200/S00158ED1V01Y200811HLT001.

[17] B. Croft and J. Lafferty, *Language Modeling for Information Retrieval*. Spring Science and Buisness Media, 2003.

[18] C. Manning and R. Socher, *Lecture 8: Recurrent Neural Networks and Language Models*. Standford University School of Engineering, 2017. Available: https://www.youtube.com/watch?v=Keqep_PKrY8t=2s

[19] V. Lavarenko and W. Croft, “Relevance-based Language Models,” in *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, NY, United States, 2001, pp. 120–127.

[20] C. Zhai and J. Lafferty, “Model-based Feedback in the Language Modeling Approach to Information Retrieval,” in *Proceedings of the International Conference on Information and Knowledge Management*, Proceedings, Oct. 2001, doi: 10.1145/502585.502654.

[21] V. Raghavan and S. Wong, “A critical analysis of vector space model in information retrieval,” *Journal of the American Society for Information Science*, vol. 37, pp. 279–287, Sep. 1986, doi: 10.1002/(SICI)1097-4571(198609)37:5¡279::AID-ASI1¿3.0.CO;2-Q.

[22] G. Salton and D. Harman, “Information Retrieval,” *Encyclopedia of Computer Science*. John Wiley and Sons Ltd., Chichester, UK, pp. 858–863, 2003.

[23] E. Greengrass, “Information Retrieval: A Survey,” Jan. 2001.

[24] G. Salton, E. Fox, and H. Wu, “Extended Boolean Information Retrieval,” *Communications of the ACM*, vol. 26, no. 11, 1983.

[25] V. Lavrenko, V. Lavarenko, *IR3.2 Overview of the vector space model*. 2015. Available: https://www.youtube.com/watch?v=19H0oWSttRst=1s

[26] V. Lavarenko, *IR3.3 Query and document vectors*,. 2015. Available: https://www.youtube.com/watch?v=dkPZXMonTLAt=1s

[27] K. Jones, “A Statistical Interpretation of Term Specificity in Retrieval,” *Journal of Documentation*, vol. 60, pp. 493–502, Jan. 2004, doi: 10.1108/00220410410560573.

[28] M. Jockers, “The LDA Buffet: A Topic Modeling Fable.” http://www.matthewjockers.net/macroanalysisbook/lda/ (accessed May 13, 2020).

[29] ISO/TC 37/SC 1, “Terminology work — Vocabulary — Part 1: Theory and application,” in *ISO 1087-1:2000*, 2000.

[30] A. Vaswani et al., “Attention Is All You Need,” in *Proceedings of Advances in Neural Information Processing Systems*, 2017.

[31] NLPL models: http://vectors.nlpl.eu/repository/

[32] M. Kusner, Y. Sun, N. I. Kolkin, and K. Weinberger, “From word embeddings to document distances,” in *Proceedings of the 32nd International Conference on Machine Learning*, ICML 2015, pp. 957–966, Jan. 2015.

[33] Project Gutenberg. Available: https://www.gutenberg.org/

[34] SpaCy. Documentation: https://spacy.io/

Appendix

A. Definitions of Categorical Tags

- ** **DEFINES** - The segment describes characteristics (3.2.4) of the concept. These characteristics are essential to a proper understanding a concept.
  Example: The segment ”The fragile offshoot issuing from the union of history and fiction is the assignment to an individual or a community of a specific identity that we can call their narrative identity.” DEFINES the concept of narrative identity.
- ** **SUB-CONCEPT** - The segment describes characteristics of a subordinate concept (3.2.14) of the concept.
  Example: The symbolism of evil is a sub-concept of the concept symbolism; narrated time is a sub-concept of the concept of time; configuration is a sub-concept of threefold mimesis.
- ** **RELATES TO** - The segment describes an associative relation (3.2.23) between the concept and another concept. Differs from sub-concepts in that the concepts do not have a hierarchical relationship.

Example: The segment ”This connection between self-constancy and narrative identity confirms one of my oldest convictions, namely, that the self of self-knowledge is not the egotistical and narcissistic ego whose hypocrisy and naivete the hermeneutics of suspicion have denounced, along with its aspects of an ideological superstructure and infantile and neurotic archaism.” RELATES TO the concept of narrative identity.

- ** **NOT RELATED** - The segment is not related at all with the concept.

  Example: The segment ”Our comparison between analytic working-through and the work of the historian facilitates the transition from our first to our second example. This is borrowed from the history of a particular community, biblical Israel. This example is especially applicable because no other people has been so overwhelmingly impassioned by the narratives it has told about itself.” is NOT RELATED to the concept of narrative identity.