AUTOMATIC METHOD FOR GENERATION OF SEMEME KNOWLEDGE BASES FROM MACHINE READABLE DICTIONARIES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE
OF BILKENT UNIVERSITY
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR
THE DEGREE OF
MASTER OF SCIENCE
IN
ELECTRICAL AND ELECTRONICS ENGINEERING

By
Ömer Musa Battal
September 2023

AUTOMATIC METHOD FOR GENERATION OF SEMEME KNOWLEDGE BASES FROM MACHINE READABLE DICTIONARIES

By Ömer Musa Battal

September 2023

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Aykut Koç(Advisor)

Tolga Çukur

Engin Erzin

Approved for the Graduate School of Engineering and Science:

Orhan Arıkan
Director of the Graduate School

ABSTRACT

AUTOMATIC METHOD FOR GENERATION OF SEMEME KNOWLEDGE BASES FROM MACHINE READABLE DICTIONARIES

Ömer Musa Battal
M.S. in Electrical and Electronics Engineering
Advisor: Aykut Koç
September 2023

The minimal semantic units of natural languages are defined as sememes. Sememe Knowledge Bases (SKBs) are organized word collections annotated with appropriate sememes. As external knowledge bases, SKBs have successful applications in multiple high-level language processing tasks. However, the construction of mainstream SKBs is performed by linguistic experts over extended periods, which restricts their prevalent usage. We present MRD4SKB as an automatic SKB generation method from readily available Machine Readable Dictionaries (MRDs). Construction of MRDs is more straightforward than SKBs, and many prominent MRDs are present in various forms. Consequently, the presented MRD4SKB is viable as a fast, flexible, and extendable method for SKB construction. Several variants of MRD4SKB, based on matrix factorization and topic modeling, are proposed to generate SKBs automatically. The performance of the automatically generated SKBs is evaluated and compared with that of other SKBs, which are constructed manually or semi-manually.

Keywords: Sememe, machine readable dictionary, sememe knowledge base, machine learning.

ÖZET

MAKİNE OKUNABİLİR SÖZLÜKLERDEN SEMEME BİLGİ TABANI ÜRETİMİ İÇİN OTOMATİK YÖNTEM

Ömer Musa Battal
Elektrik Elektronik Mühendisliği, Yüksek Lisans
Tez Danışmanı: Aykut Koç
Eylül 2023

Doğal dillerin asgari semantik birimleri sememeler olarak tanımlanır. Sememe Veri Tabanları (SVT), uygun sememeler ile şerh düşülmüş düzenli kelime derlemeleridir. Harici veri tabanları olarak SVT'ler, birçok yüksek seviye dil işleme görevinde başarılı uygulamalara sahiptir. Buna karşın, ana akım SVT'lerin inşası dilbilim uzmanları tarafından uzun süreler içinde icra edilir, ki bu da onların yaygın kullanımını kısıtlar. MRD4SKB'yi kolayca bulunabilen Makine Okunabilir Sözlüklerden (MOS) otomatik SVT üretimi için bir yöntem olarak sunuyoruz. MOS'ların inşası SVT'lerinkine kıyasla daha dolambaçsızdır ve farklı formatlarda birçok seçkin MOS bulunmaktadır. Bu nedenle, sunulan MRD4SKB hızlı, esnek ve genişletilebilir bir SVT inşa yöntemi olarak uygulanabilirdir. Otomatik SVT üretimi için MRD4SKB'nin matrix faktörizasyonu ve konu modellemesine dayalı çeşitli biçimleri önerilmektedir. Otomatik olarak üretilen SVT'lerin performansları değerlendirilmekte ve diğer elle yahut yarı-elle inşa edilmiş SVT'lerinki ile karşılaştırılmaktadır.

Anahtar sözcükler: Sememe, makine okunabilir sözlük, sememe bilgi tabanı, makine öğrenmesi.

Acknowledgement

I would like to thank my advisor Dr. Aykut Koç for his guidance throughout my graduate education and research.

I would like to thank the thesis committee members for reviewing my thesis.

I would like to express my gratitude to my family and friends for their support during my study.

Finally, I would like to thank The Scientific and Technological Research Council of Turkey (TÜBİTAK) for providing financial support.

Contents

List of Figures

List of Tables

List of Abbreviations

NLP	Natural Language Processing
SKB	Sememe Knowledge Base
MRD	Machine Readable Dictionary
RNN	Recurrent Neural Network
LSTM	Long Short-Term Memory
GRU	Gated Recurrent Unit
ASGD	Averaged Stochastic Gradient Descent
AWD	ASGD Weight-Dropped
SDLM	Sememe-Driven Language Model
WSD	Word Sense Disambiguation
CDV	Controlled Defining Vocabulary
MAP	Mean Average Precision
NMF	Non-Negative Matrix Factorization
LSA	Latent Semantic Analysis
pLSA	Probabilistic LSA
LDA	Latent Dirichlet Allocation

PTB	Penn Treebank
CCSA	Consistency Check with Sememe Annotations
LM	Language Modeling
NLI	Natural Language Inference
SA	Sentiment Analysis
TAA	Textual Adversarial Attacking
ASR	Attack Success Rate
IGE	Increase in Grammatical Error
PPL	Perplexity

Chapter 1

Introduction

The most minor elements of natural languages that can stand by themselves are defined as words. However, words are not the most minor indivisible semantic units. ‘Sememes’ are defined as the most basic semantic units of natural languages [3]. To demonstrate this concept with some examples, the word `school` can be regarded as a composition of the meanings of `education` and `building`. Similarly, the word `hospital` can be regarded as a composition of the meanings of `medicine` and `building`. The terms `education`, `medicine`, and `building` are considered sememes in the given examples, and their combinations are used to annotate the words `school` and `hospital`. The sememe annotation method in the given examples is relatively simple but still practical. More complex and hierarchical sememe annotation methods exist, which are used in specific contexts for varying intents. The prominent example of this is the HowNet [4].

The usefulness of sememes has been demonstrated in various natural language processing (NLP) tasks [5]. An assortment of these works will be given as follows.

- *word similarity computation:* Sememes have been utilized in Chinese lexical fusion, which can be considered a specialized word similarity task [6]. Chinese lexical fusion occurs when a fusion form with the same meaning duplicates a pair of words in a sentence. This fusion form generally consists of rarer words, which hinders downstream NLP tasks like reading comprehension, summarization, and machine translation. Sememe knowledge is used to capture the semantic similarity between the original and fusion forms of these word pairs in a sentence to avoid degradation in downstream task performance.

- *word representation learning:* Sememes have been used in the skip-gram framework to learn word representations [7]. Word sememes are utilized to accurately capture the exact meaning of a word within a specific context through an attention mechanism.

- *sentiment analysis:* Sememes have been employed in the sentiment analysis of Chinese online social reviews [8]. First, a topic model is trained on the review dataset. The trained topic model then identifies the topics of a document, and sememes are used to classify these topics based on their sentiment polarity.

- *definition generation:* Sememes have been utilized in automatically generating definitions for specific words. In one of the previous works, an encoder-decoder framework is used [9]. The encoder maps words and sememes into a sequence of continuous representations. The decoder then attends to the output of the encoder and generates the definitions. In another work, meanings of words are explicitly decomposed into their semantic components through sememes, and these decompositions are then used as discrete latent variables for a definition generation model [10].

- *lexical simplification:* Replacing complex words in a sentence with simpler alternatives of equivalent meaning is defined as lexical simplification, and sememes have been used for this purpose [11]. Two words are considered semantically equivalent if sememe annotations of one of their senses are the same. This approach is used for finding complex words to substitute with simpler words of identical meaning.

- *lexicon expansion:* Sememes have been used for automatically expanding the lexicon of a popular word counting software tool named Linguistic Inquiry and Word Count (LIWC) [12]. Sememe information is used for distinguishing word meanings to address the polysemy and indistinctness problems encountered during lexicon expansion. An attention mechanism is also employed to assign weights to the sememes to utilize the sememe information better.

- *text classification:* Sememes have been employed in Chinese question classification task [13]. The questions are generally classified based on what kind of answer is given to them, which is conceptualized as Question Focus Words (QFW). The sememes of the QFWs are used for performing finer classifications of Chinese questions.

- *language modeling:* Sememe-Driven Language Model (SDLM) has been proposed for language modeling task [14]. In SDLM, the next word is predicted using an estimate of sememe distribution given the textual context. Each estimated sememe is regarded as a semantic expert, and the most probable senses, and their corresponding word, are jointly identified by these semantic experts to predict the next word in the sequence.

- *sarcasm detection:* Sememe and Auxiliary Enhanced Attention Model (SAAG) has been presented for Chinese sarcasm detection [15]. Sememe knowledge is used for enhancing the representation learning of Chinese words at the word level. The representation of text expression is then progressively and dynamically constructed based on the sememe-enhanced word representations to detect sarcasm.

Integration of linguistic knowledge into recurrent neural networks (RNNs) has been a topic of research [16–19]. This line of work opened the way to incorporating sememe information into neural network models such as RNNs [20] and transformers [21] in order to increase their performance on NLP tasks.

Every sense of each word can be expressed as a combination of suitable sememes, provided that a well-constructed predefined sememe set exists [3]. However, the construction of this predefined set of sememes is a challenging task. Moreover, even after the base sememe set is constructed, annotating words with proper sememes is non-trivial. These tasks have formerly been performed manually by linguistic experts over the years [4,22]. Involving groups of linguistic experts is not uncommon in the field of NLP [23]. However, this strategy is not always optimal. Creation of manual datasets for NLP tasks such as text alignment is not trivial, even for experts [24]. Furthermore, the manually prepared datasets usually end up being small-sized and domain-specific. Another method for dataset construction is by crowd-sourcing. Nevertheless, producing high-quality data for complex tasks may require educating the crowd workers, which can be costly [23]. When these facts are taken into account, it comes as no surprise that automatic or semi-automatic dataset construction methods have been used in several NLP tasks [25–28]. The task of sememe annotation has been automated to various extents. Annotating words with sememes from a predefined set is called lexical sememe prediction, and there are numerous methods of automating this task present in the literature [1,29–37]. Using a controlled defining vocabulary as an initial sememe set and then performing sememe annotations from them to create an SKB is also introduced [38]. Despite this, fully automatic initial sememe set generation using regular MRDs remains a feat yet to be achieved.

We present MRD4SKB as a method of using a machine-readable dictionary (MRD) to generate a set of predefined sememes and the corresponding sememe annotations for the words in the same MRD to construct a sememe knowledge base (SKB) that represents the target language optimally. Utilizing available well-established MRDs to construct SKBs automatically eliminates the need for complicated manual work that requires linguistic expertise, which can stimulate further sememe-related NLP research. Using our method to construct the base set of predefined sememes happens to perform the lexical sememe prediction task as a by-product, which is crucial for establishing an SKB.

A basic description of MRD4SKB can be given as a set of consecutive steps. First, a term-document matrix is constructed from the MRD of interest using the preprocessing methods explained in Section 3.1. Then, appropriate dimensionality reductions are applied to this term-document matrix as column reductions. Two main approaches are proposed to perform matrix dimensionality reduction. In the first approach, a customized Kronecker-product-based matrix reduction method is used, details of which are explained in Section 3.2. In the second approach, We utilize topic modeling methods such as Non-Negative Matrix Factorization (NMF), Latent semantic analysis (LSA), and Latent Dirichlet Allocation (LDA), details of which are described in Section 3.3. The rest of the thesis is organized in the following manner. Further information on SKBs and previous work regarding them is presented in Chapter 2. We explain the details of our proposed MRD4SKB method in Chapter 3. Experimental results and evaluations are presented in Chapter 4. The thesis concludes in Chapter 5.

Chapter 2

Sememe Knowledge Bases (SKBs)

SKBs are knowledge bases (KBs) principally constructed to comprise sememe annotations of words. The sememes are chosen from a predefined sememe set, which is also a part of the SKB. This predefined set of sememes can be regarded as *the periodic table* for a natural language, or the indivisible atoms of meaning, as an analogy. An SKB strives to define the words in a vocabulary by compounding these sememes properly, using weighted and hierarchical means. HowNet is the most famous SKB in the literature [4]. HowNet is the result of the manual efforts of linguistic experts over the years. Fig. 2.1 displays a sample entry from the HowNet.

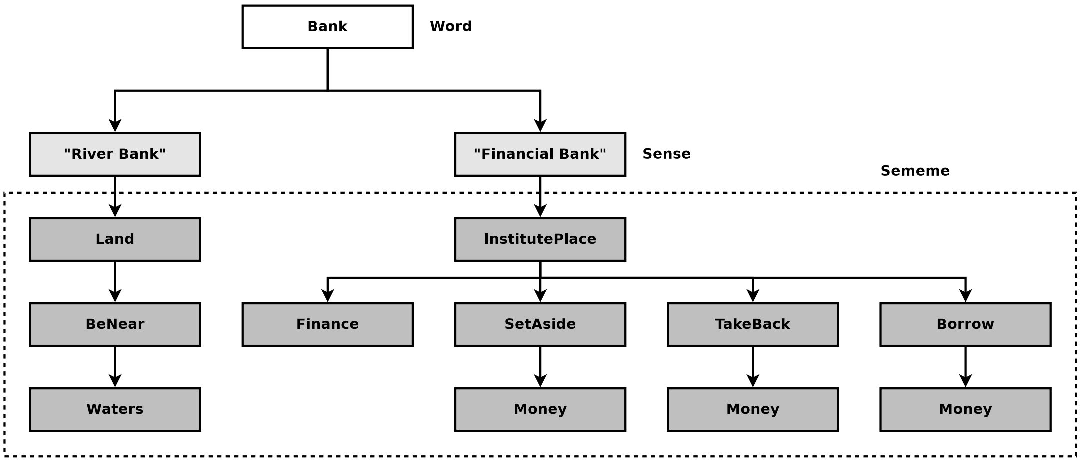

Figure 2.1: An example of a word annotated with HowNet [1].

The word `bank` is annotated with sememes in the given manual sememe annotation entry example. Two senses of the word `bank` are annotated: “river bank” and “financial bank”. Each of these senses is manually annotated with appropriate sememes selected from the base sememe set of HowNet, which consists of about $2,500$ sememes. The sememe set of HowNet is manually determined by extracting, analyzing, merging, and filtering the semantics of thousands of Chinese characters. In HowNet, sememe annotations are performed hierarchically, which contains further information on the meaning of the annotated senses. HowNet was initially designed and constructed in the 1990s and was published in 1999 [22]. Moreover, its original authors have kept it frequently updated since its publication through manual efforts.

SKBs are utilized to enhance the performances of machine learning models in numerous NLP tasks [5]. Some of these can be listed as follows.

- SKBs have been utilized in improving the sequence modeling ability of RNNs [20]. Three sememe incorporation methods are designed and employed in typical RNNs such as LSTM, GRU, and their bidirectional variants.

- SKBs have been used in constructing sememe enhanced transformer models [20]. Sememe knowledge is incorporated into transformer models through sememe embeddings, auxiliary sememe prediction task, and their hybrid combination. It has been shown that sememe incorporation substantially improved the model’s robustness against adversarial examples.

- SKB-based semantic relatedness measures have been applied in word sense disambiguation (WSD) [39]. Concept Relevance Calculator (CRC) is provided as part of HowNet and is used to construct sense colonies or bags of concepts. Then, a machine learning tool named Conditional Random Fields (CRFs) is used for WSD, utilizing the constructed sense colonies.

- SKBs have been utilized in annotating information structures in Chinese texts [40]. An information structure is constructed as a combination of SKB definitions and dependency relations. Incorporation of SKB information to corpora is intended to contribute to improving text understanding and machine translation.

- An unsupervised neural framework has been proposed that leverages sememes for aspect extraction [41]. A framework analogous to an auto-encoder is used, where sememes are leveraged to form input sentence representations through various attention mechanisms.

- SKBs have been beneficial for modeling semantic compositionality (SC), which refers to the concept of decomposing the meaning of a complex linguistic unit into the meanings of its constituents [42]. Sememe information is integrated into SC models for learning representations of multi-word expressions that better correlate with human judgment.

- SKBs have been used in reverse dictionary models [43]. A reverse dictionary takes the description of a word as input and outputs the target word, which is the opposite task of a regular forward dictionary. Sememe annotations obtained from SKBs have been used to design a sememe predictor, which is utilized for predicting the target word from a given input description.

In addition, SKBs have been utilized in language modeling [14], quantifying word semantic similarities [44], and lexical fusion recognition [6]. Furthermore, a basis for the theoretical implications of semantic compositionality, an important NLP topic, can be provided through SKBs [45].

Advanced expertise in linguistics and laborious effort are required for building SKBs, as seen in the case of HowNet. Extending an existing SKB with sememe annotations of previously unannotated words is called lexical sememe prediction. The task of lexical sememe prediction is studied through numerous approaches [1, 29, 32–37]. Some of these methods utilize dictionaries as well [30, 31]. To construct an SKB, rather than extending an existing one, the EDSKB approach was previously proposed [38]. EDSKB uses a manually crafted dictionary named *Controlled Defining Vocabulary (CDV)* as the initial sememe set. CDV consists of a well-chosen list of words used to construct all definitions in the broader dictionary, which is deemed suitable for being an initial set of sememes. Preprocessing and tokenizing steps are applied to all dictionary definitions, and tokens not contained in the predefined sememe set are removed. The remaining tokens are then taken as the sememe annotations of the words in the dictionary [38]. In order to further reduce the number of annotated sememes per term, the authors of EDSKB additionally introduce dependency parsing to create alternative SKBs [46]. Although an SKB can be constructed by the EDKSB method, reliance on a manually prepared CDV as the initial sememe set limits its utility. Moreover, it leaves the automatic generation of the initial sememe set as an open problem. It is, therefore, desirable to generate both the predefined sememe set and the corresponding sememe annotations through a single automatic method. In theory, such a method can choose the optimal set of initial sememes that would represent the semantics of the language. Additionally, the automatic building of the initial sememe set creates the opportunity to use more extensive dictionaries for which a manually constructed CDV does not exist. With this motivation, we propose the MRD4SKB method as an approach that generates the initial sememe set and performs the necessary sememe annotations to build an SKB automatically in the next section.

Chapter 3

Automatic Sememe Knowledge Base Generation

MRD4SKB is a computational method for generating SKBs from any machine-readable dictionary (MRD) in a fully automatic manner. An MRD database with machine-readable formatting that contains words, their definitions, and possibly other related information [47]. Since humans rely on dictionaries to learn, use, and study languages, Constructing dictionaries is almost always necessary for any natural language. Hence, significant effort is spent on building MRDs, many provided and published online. WordNet [48] and Wiktionary [49] are among the most popular and established MRDs. In addition to manual dictionary creation, automatic construction of dictionaries [50] and reverse dictionaries [51] are both researched in the NLP field. Various NLP tasks utilize MRDs, such as language model enhancement [52]. The size and content of MRDs can vary considerably, so selecting an MRD should be appropriately made while considering the requirements of the particular application.

Dictionary construction is much more straightforward compared to coming up with a predefined set of sememes and annotating words with them. Sememes are restrictive, whereas a dictionary definition can use the entire vocabulary. We assume here that a comprehensive dictionary should contain most of the words necessary to encompass the semantic space of a language. Therefore, the dictionary should already contain almost all sememes within itself. The problem then remains to extract these sememes, and the matching sememe annotations, from the dictionary. We first construct a document-term matrix from the chosen MRD to base our further work. We then propose two primary methodologies to extract sememes and sememe annotations from the generated document-term matrix in separate sections.

3.1 Construction of MRD Document-Term Matrix

An essential part of this work is to model an MRD with a document-term matrix. A document-term matrix is a matrix of the frequency of terms within a collection of documents. In the traditional definition, the matrix rows correspond to the documents in the collection, and the columns represent the individual terms. The matrix entries specify the occurrence of a specific term in the specified document. The transposition of this matrix is appropriately called a term-document matrix, in which the row and column definitions are interchanged. We will be using the document-term matrix format in this work. It should come as no surprise that an MRD can be considered a collection of documents. In this case, the documents are the definitions of individual words provided by the dictionary. Each document is titled with the word it defines, which is itself another term. Due to polysemy, a word can have multiple senses, meaning it can also have multiple documents. We performed word sense disambiguation (WSD) on the MRD by applying the Lesk algorithm to avoid two documents having the same word as their title [53]. As a result, each word in the MRD is labeled by a sense number, corresponding to a sense of that particular word in the MRD itself. We then use these distinguishable sense-labeled words as the terms of the MRD. A mathematical representation of the MRD can then be presented as follows:

$$\begin{aligned} t_1 &: [t_{1,1}, t_{1,2}, \cdots, t_{1,s_{t_1}}] = d_1 \\ &\vdots \\ t_i &: [t_{i,1}, t_{i,2}, \cdots, t_{i,s_{t_i}}] = d_i \\ &\vdots \\ t_m &: [t_{m,1}, t_{m,2}, \cdots, t_{m,s_{t_m}}] = d_m, \end{aligned} \tag{3.1}$$

where m is the total number of terms that have definitions in the MRD, and t_i is the i'th term, which has a total number of s_{t_i} words in its definition document d_i. After this construction, tokenization, lemmatization, and stop-word removal steps are applied to documents d_i in order to produce a preprocessed $(m \times n)$ dimensional document-term matrix $\mathbf{D}$, where n is the total number of different terms used in all definitions of the MRD combined. The preprocessing steps utilize the algorithms and modules from the SpaCy software library [54].

The i^{th} row of $\mathbf{D}$, which will be denoted as $\mathbf{D}_{(i,*)}$, corresponds to a term t'_i that *has* a definition in the MRD, and will be called the *definition vector* of t'_i. The j^{th} column of $\mathbf{D}$, which will be denoted as $\mathbf{D}_{(*,j)}$, corresponds to a term t''_j that is *used* in at least one definition in the MRD, and will be called the *occurrence vector* of t''_j. The rows and columns of $\mathbf{D}$, which have the following properties:

$$\begin{aligned} \mathbf{D}_{(i,*)} &= \mathbf{0}, \\ \mathbf{D}_{(*,j)} &= \mathbf{0}, \end{aligned} \tag{3.2}$$

are removed from $\mathbf{D}$ as part of pre-processing. These correspond to a term t'_i with the null definition vector and a term t''_j with the null occurrence vector.

Let $\mathbf{t}'$ and $\mathbf{t}''$ denote the lexically ordered sets of all t'_i's and all t''_j's of $\mathbf{D}$, respectively. Then let the lexically ordered set $\mathbf{t}$ be defined as their intersection as:

$$\mathbf{t} = \mathbf{t}' \cap \mathbf{t}''. \tag{3.3}$$

Note that the definition of $\mathbf{t}$ above allows the following to hold:

$$t'_i = t''_j = t_k,\ \exists!\{i, j, k\},\ \forall k, \tag{3.4}$$

which means that, given a column term t''_j, if $t''_j \in \mathbf{t}$ is satisfied, then $t_k = t''_j$ can instead be used, and if $t''_j \notin \mathbf{t}$, then no such t_k exists. Similar also holds for row terms t'_i.

We introduce the following matrix indexing notation, which will be useful later:

$$\begin{aligned} \mathbf{D}_{(i,*)} &= \mathbf{D}_{(t'_i,*)}, \\ \mathbf{D}_{(*,j)} &= \mathbf{D}_{(*,t''_j)}, \\ \mathbf{D}_{(i,j)} &= \mathbf{D}_{(t'_i,t''_j)}, \end{aligned} \tag{3.5}$$

where matrix $\mathbf{D}$ is not indexed by integers, but by the corresponding terms.

The automatic SKB generation algorithm starts with the following initial condition: the terms t''_j from columns of the matrix $\mathbf{D}$ are set as sememes, and $\mathbf{D}_{t'_i,*}$ is the sememe annotation row vector for the term t'_i. However, the direct application of this proposition has an obvious issue. The width of the original document-term matrix $\mathbf{D}$ is impractical as an SKB because its width is excessive. A wide SKB matrix produces many sememes with little or too specific semantic content. Annotating terms from a large sememe pool like this is effectively useless. In other words, column terms of the initial matrix can not be regarded as sememes. Ideally, a matrix $\mathbf{M}$ should be obtained from $\mathbf{D}$, which should be as narrow as possible but not narrower. Even in the case of manual extraction by linguistic experts, the questions of “What is the optimal number of sememes?" and “What are these ultimate sememes?" are open problems. With this in consideration, the semantic content of sememes should not be too specific, but the composition of sememes should allow for building the semantic representations of specific terms. Hence, a dimensionality reduction process should be applied to matrix $\mathbf{D}$ to obtain a narrower matrix $\mathbf{M}$.

3.2 Kronecker-Product Based Matrix Reduction

Various dimensionality reduction methods exist, such as PCA [55] and non-negative matrix factorization [56]. These methods achieve dimensionality reduction by creating a new feature space. Determining what the dimensions of this new feature space correspond to requires an interpretation step. In order to avoid this interpretation step, we propose an iterative matrix reduction method customized to suit our purposes in this application. The state of the document-term matrix $\mathbf{M}$ at l^{th} iteration will be denoted as $\mathbf{M}^{(l)}$, and the dimension of the matrix will be denoted as $(m^{(l)}, n^{(l)})$, where l will start from 0. Note that $\mathbf{M}^{(0)} = \mathbf{D}$, which is the original MRD document-term matrix.

In our dimensionality reduction step, we remove the columns with the least total frequency from the matrix $\mathbf{M}$, thereby narrowing it. To this end, we use the following column selection procedure:

$$\begin{aligned} \mathbf{1} \cdot \mathbf{M}^{(l)} &= \mathbf{C}^{(l)}, \\ \min_i(\mathbf{C}^{(l)}_{(1,i)}) &= c^{(l)}, \\ \arg\min_i(\mathbf{C}^{(l)}_{(1,i)}) &= i^{(l)}_{min} \rightarrow t''_{i^{(l)}_{min}}, \end{aligned} \tag{3.6}$$

where $\mathbf{C}$ is the $(1, n^{(l)})$ column sum vector, $\mathbf{1}$ is the $(1, m^{(l)})$ matrix of all ones, $c^{(l)}$ is the minimum column sum, and $t''_{i^{(l)}_{min}}$ is the term or terms corresponding to this minimum column sum. The $t''_{i^{(l)}_{min}}$ are the least frequently used terms over all definitions, so their semantic contents are assumed to be either not very significant or too specific to be considered as a sememe. Thus, we must remove them from the list of possible sememe candidates. However, the direct removal of these columns has a detrimental effect on the semantic content of the remaining part of $\mathbf{M}$. To preserve the semantic content through information diffusion, if the following condition is met:

$$t''_{i^{(l)}_{min}} \in \mathbf{t} \Rightarrow t''_{i^{(l)}_{min}} = t_k, \ \exists! k, \tag{3.7}$$

the following update step is added to our proposed method:

$$\begin{aligned}\mathbf{\Delta_l} =& \mu M_{t'_i, t''_{i^{(l)}_{min}}} \mathbf{M}_{t''_j,*} \\ \mathbf{M}_{t'_i,*} \leftarrow& \mathbf{M}_{t'_i,*} + \mathbf{\Delta_l} \ , \ \forall t'_i,\end{aligned} \tag{3.8}$$

where μ is a semantic relation factor to be described later, and $\mathbf{\Delta_l}$ is the delta element of our update step. The update step in Eq. 3.8 is performed right before the reduction step in each iteration. The components of the delta element $\mathbf{\Delta_l}$ in the update step are justified as follows, where we will use $t_j = t''_{i^{(l)}_{min}}$ to simplify notation. M_{t_i,t_j} denotes the semantic relation between term t_i and sememe candidate t_j. $\mathbf{M}_{t_j,*}$ is interpreted as the semantic compositionality vector of the sememe candidate t_j, which is originally the definition vector obtained from the preprocessed MRD document-term matrix. Note that this sememe candidate is a failed one; it will be removed and not considered in the following iterations.

Finally, μ is a semantic relation factor, taking values in the range $[0, 1]$. The purpose of this factor is to retain part of the lost semantic content by removing a sememe candidate column from the SKB matrix $\mathbf{M}$. It achieves this by diffusing the information within this sememe candidate to other definition rows containing it. For instance, if the sememe candidate `hospital` were removed, other definition rows containing this sememe candidate would lose information. Adding the definition row vector of the term `hospital` to these definition row vectors with a factor of μ diffuses the information contained in this sememe candidate to other terms and reduces the loss of semantic content in the final SKB. Higher values of μ would result in a higher rate of information diffusion, but it would risk suppressing the information content from the original sememes of the terms. Therefore, the value of μ should be fine-tuned.

A useful feature of this update step is that Eq. 3.8 turns out to be equivalent to a simple Kronecker product when applied to the entire matrix $\mathbf{M}$ for a single sememe candidate t_j, as the following:

$$\mathbf{M}'^{(l)} = \begin{cases} \mathbf{M}^{(l)} + \mu(\mathbf{M}_{*,t_k} \otimes \mathbf{M}_{t_k,*}), & \text{if Eq. 3.7 holds,} \\ \mathbf{M}^{(l)}, & \text{otherwise.} \end{cases} \tag{3.9}$$

In either case, the column reduction step is performed as follows to obtain the next iteration:

$$\mathbf{M}^{(l+1)} = \mathbf{M'}^{(l)} \cdot \mathbf{J}^{T}_{(i^{(l)}_{min}, n^{(l)})}, \tag{3.10}$$

where $\mathbf{J}_{(i,n)}$ is defined as a $(n-1) \times n$ matrix that comes from removing the i^{th} row from the identity matrix $\mathbf{I}_n$. The advantage of the Kronecker product is its computational efficiency when applied to sparse matrices.

Decreasing the matrix size and increasing its density is achieved by the iterative application of the Kronecker-based column reduction explained above. However, if no stopping condition is specified, the reduction algorithm will conclude in an SKB matrix $\mathbf{M}$ with 0 columns. This is not desirable since all information content will be lost at that point. Two stopping conditions are proposed to avoid this issue. The first stopping condition is a threshold of minimum column sums of $\mathbf{M}$, the value of which is tuned through experiments. This method ensures that all remaining final sememe candidates are lower-bounded in terms of their semantic content over all defined terms in the dictionary. The second stopping condition is the maximum number of final sememes. Heuristically, existing manually crafted SKBs such as HowNet use an approach established on the semantic representation capabilities of the finite set of commonly used Chinese characters. The average Chinese speaker needs to know around $2,000$ characters to be recognized as fluent, and HowNet also contains around that many sememes in their hand-made SKB [4]. Considering this, the importance of the total number of sememes in an SKB is recognized. Thus, we use the final number of sememes as a parameter to be tuned through our experiments. Definitions of our two proposed stopping conditions are given below mathematically:

$$\begin{aligned} c^{(l)} &< C, \\ n^{(l)} &< N, \end{aligned} \tag{3.11}$$

where C and N are the threshold values for their corresponding variables. We use a single sememe prediction experiment to determine values for these thresholds as a validator. The values which maximize the validation performance in the selected validator task are subsequently fixed. We then report the results using these optimized hyperparameter values in other intrinsic and extrinsic experiments.

The MRD4SKB variant that utilizes Kronecker product-based matrix reduction will hereafter be referred to as the MRD4SKB_KP method. An algorithmic description of it can be summarized in Algorithm 1.

Algorithm 1: MRD4SKB_KP algorithm

```
Input  : ($n \times m$) document-term matrix of MRD, $\mathbf{D}$;
         discount factor $\mu$;
         maximum number of sememes $N$
Output : ($n \times s$) term-sememe matrix of SKB, $\mathbf{M}$
$\mathbf{M} = \mathbf{D}$
colsums = colsum($\mathbf{M}$)
n = width($\mathbf{M}$)
while $n > N$ do
    colsums = colsum($\mathbf{M}$)
    minterms = argmin(colsums)
    deltamtx = 0
    for minterm in minterms do
        deltamtx += kron($\mathbf{M}$.col[minterm],$\mathbf{M}$.row[minterm])
    end
    $\mathbf{M}$ += $\mu \times$ deltamtx
    M.drop($\mathbf{M}$.col[minterms])
    n = width($\mathbf{M}$)
end
return $\mathbf{M}$
```

In the MRD4SKB_KP algorithm, the semantic relations between the terms are captured by the Kronecker product. The strength of the semantic relation between the terms is adjusted by the discount factor μ. The number of sememes left in the final SKB matrix **M** is controlled by the final parameter N.

The matrix **M** that results from applying the MRD4SKB_KP algorithm can be regarded as a word vector space, where each vector space dimension corresponds to a known sememe. The dimensions are primarily uninterpretable in a common word vector space, which hinders its explainability. Furthermore, a connection between the matrix **M** and common word embeddings can be established by introducing the sememe embedding concept. In Sememe Prediction with Aggregated Sememe Embeddings (SPASE) method [29], a sememe annotation matrix relates the word embedding vectors and sememe embedding vectors, as given below:

$$\mathbf{w}_i = \sum_{s_j \in S_{w_2}} \mathbf{M}_{ij} \cdot \mathbf{s}_j, \tag{3.12}$$

where S_{w_i} is the sememe set of the word w_i, and $\mathbf{M}_{ij}$ represents the weight of sememe s_j for word w_i. SPASE attempts to decompose a word embedding matrix $\mathbf{W}$ into sememe annotation matrix $\mathbf{M}$ and sememe embedding matrix $\mathbf{S}$, with word embeddings that are pre-trained and fixed during training, which can also be written as $\mathbf{W} = \mathbf{M} \times \mathbf{S}$. In SPASE, the matrix $\mathbf{M}$ is taken from a manually constructed SKB. Our method automatically constructs the SKB and the sememe annotation matrix $\mathbf{M}$. Although sememe embeddings are beyond the scope of this manuscript, they need to be noted as a future topic of interest.

In addition to being fully automatic, our proposed method has another advantage. Current methods in the literature that embed sememes to RNNs use binary word sememe annotations, like in the case of SPASE. This is because most SKBs contain binary sememe annotations for words, where each word is annotated with a composition of sememes with binary weights. In contrast, the SKB matrix $\mathbf{M}$ automatically constructed with our proposed method does not necessarily produce binary sememe annotation weights because of the semantic relation factor μ. Therefore, the generated sememe annotations can be directly used to decompose a word into its sememes as a weighted sum. This is useful because, for a particular word, some sememes may be more semantically relevant than others. For example, recall the word `hospital` that was given as an example in the Introduction. To keep the example relevant, annotate this word by the hypothetical sememes `medicine`, `building`, `doctor`, and `patient`. If a binary annotation approach is used, no distinction between these annotated sememes can be made. However, it can be argued that the sememes `medicine` and `building` are more important semantic components of the word `hospital` compared to the other two. Therefore, assigning a weight to each of these sememes, used as annotations for a particular word, is helpful, and our method can readily perform it. Furthermore, if a binary annotation is necessary for specific tasks, a basic thresholding can be easily performed as follows:

$$S_{t_i} = \{t_j | M_{t_i,t_j} \geq k, \forall t_j\}, \tag{3.13}$$

where S_{t_i} is the sememe set of the term t_i, and k is the binary sememe annotation threshold, taking non-negative real values. $k = 0$ case keeps all sememe annotations, and increasing this value reduces the annotated sememe counts in general. In the experiments that require binary sememe annotation, we add this k value as a hyperparameter to be tuned in the experimental setup.

3.3 Topic Modeling Based Matrix Reduction

In the previous section, we explained a Kronecker Product-based method for automatically generating an SKB from a term-document matrix obtained from a preprocessed MRD. The presence of a term-document matrix and recognizing an analogy between sememes and topics made us consider utilizing topic modeling methods for generating SKBs from MRDs. We present our alternative topic modeling-based MRD4SKB method variants in this section.

Topic modeling is a statistical tool for extracting latent variables from large datasets [2,57]. It is frequently used in text-mining applications to discover hidden semantic structures of a corpus of collected documents. Studying the development of ideas in a scientific field could be given as an example application [58]. Topic modeling is based on the hypotheses that documents are about one or more topics (statistical mixture hypothesis) and that documents about similar topics use similar terms (distributional hypothesis). A topic model captures this intuition in a mathematical framework by utilizing term-document statistics of the corpus. This framework is visualized in Fig. 3.1.

In topic modeling, corpus statistics are generally captured by a term-document matrix in a bag-of-words (BoW) fashion, discarding the order of the terms. The construction of a term-document matrix from a corpus (specifically, an MRD) was

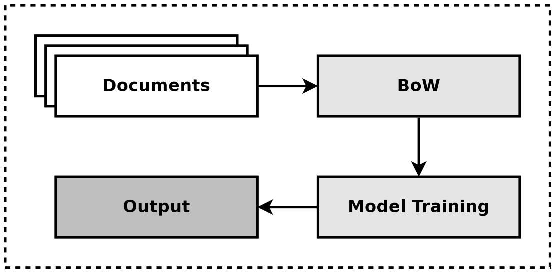

Figure 3.1: Framework of topic modeling [2].

previously discussed in Section 3.2. Tf-idf is a commonly used pre-transformation to refine the statistics represented through the term-document matrix by reweighting the matrix elements with a combination of term frequency (tf) and inverse document frequency (idf) [59]. Tf-idf computation is defined as:

$$\begin{aligned} \mathrm{tf}(t,d) &= \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}, && (3.14)\\ \mathrm{idf}(t,D) &= \log \frac{N}{1 + |\{d \in D : t \in d\}|}, && (3.15)\\ \text{tf-idf}(t,d,D) &= \mathrm{tf}(t,d) \cdot \mathrm{idf}(t,D), && (3.16) \end{aligned}$$

where t and d denote terms and documents, respectively, $f_{t,d}$ is the raw count of t within d, and N is the total number of documents in corpus D. The denominator of $\mathrm{idf}(t,D)$ is the number of documents where term t appears, and 1 is added to avoid division by zero.

After obtaining the term-document matrix and possibly performing tf-idf on it, topic modeling is mainly concerned with using this matrix as input and generating two output components by introducing topics: a term-topic matrix and a topic-document matrix. The term-topic matrix contains the weights of the corpus terms on each introduced topic, and the topic-document matrix contains the distribution of these topics on the corpus documents. Obtained topic information can then be used in various NLP tasks, such as sentiment analysis [60], text summarization [61], and text categorization [62].

In MRD4SKB, we use these topic-modeling-related matrices in their transposed forms. The preprocessed MRD matrix **D** is used as the document-term matrix.

The generated document-topic matrix is denoted as $\mathbf{M}$, and the topic-term matrix is denoted as $\mathbf{T}$, where the relation among these matrices is the following:

$$\mathbf{D} = \mathbf{MT}. \tag{3.17}$$

Note that we intentionally used the same variable $\mathbf{M}$ for the automatically generated SKB matrix and the generated document-topic matrix, as we will be using the document-topic matrix as our generated SKB in topic modeling methods. Here, documents correspond to definitions of the terms in the MRD, and topics are assumed to be the selected base sememe set of the SKB. We utilize the topic-term matrix $\mathbf{T}$ to obtain precise sememe terms for each topic. The sememe s_i corresponding to document-topic vector $\mathbf{M}_{(*,\mathbf{i})}$ and topic-term vector $\mathbf{T}_{(\mathbf{i},*)}$ is defined by:

$$s_i = t''_j \text{ where } j = \arg\max_j(\mathbf{T}_{(i,j)}), \tag{3.18}$$

where we used our notation in Eq. 3.3 for t''_j. After this procedure, the resulting labeled document-topic matrix $\mathbf{M}$ becomes the automatically generated SKB.

There are various methods of obtaining topic model outputs from term-document matrices, such as the Non-Negative Matrix Factorization (NMF), Latent Semantic Analysis (LSA), probabilistic LSA (pLSA), and Latent Dirichlet Allocation (LDA). The following subsections briefly describe how each method works and how the document-topic matrix $\mathbf{M}$ and the topic-term matrix $\mathbf{T}$ are built. Once these are clarified, we use the procedure in the previous paragraph to finalize the automatically generated SKB.

3.3.1 Non-Negative Matrix factorization (NMF)

NMF is a group of algorithms where the term-document matrix $\mathbf{D}$ is factorized into two matrices $\mathbf{M}$ and $\mathbf{T}$, where all three matrices have non-negative elements as follows [63]:

$$\mathbf{D} = \mathbf{MT}, \tag{3.19}$$

where $\mathbf{D}$ is non-negative by definition, and $\mathbf{M}$ and $\mathbf{T}$ are initialized as non-negative. Then, the following updates are performed on $\mathbf{M}$ and $\mathbf{T}$ until they stabilize, with n as the index of iteration:

$$\mathbf{T}^{n+1}_{[i,j]} \leftarrow \mathbf{T}^{n}_{[i,j]} \frac{\left(\left(\mathbf{M}^{n}\right)^{T} \mathbf{D} \right)_{[i,j]}}{\left(\left(\mathbf{M}^{n}\right)^{T} \mathbf{M}^{n} \mathbf{T}^{n} \right)_{[i,j]}}, \tag{3.20}$$

$$\mathbf{M}^{n+1}_{[i,j]} \leftarrow \mathbf{M}^{n}_{[i,j]} \frac{\left(\mathbf{D} \left(\mathbf{T}^{n+1}\right)^{T} \right)_{[i,j]}}{\left(\mathbf{M}^{n} \mathbf{T}^{n+1} \left(\mathbf{T}^{n+1}\right)^{T} \right)_{[i,j]}}. \tag{3.21}$$

In MRD4SKB, we use the matrices $\mathbf{M}$ and $\mathbf{T}$ as they were denoted before. This method will be referred to as MRD4SKB_TM_NMF.

3.3.2 Latent Semantic Analysis (LSA)

LSA method uses truncated Singular Value Decomposition (SVD) to obtain the topic outputs from the term-document matrix $\mathbf{D}$ as follows:

$$\mathbf{D} = \mathbf{M}\boldsymbol{\Sigma}\mathbf{T}^{T}, \tag{3.22}$$

where $\mathbf{M}$ and $\mathbf{T}$ are orthogonal matrices and $\boldsymbol{\Sigma}$ is a diagonal matrix. The following gives the derivation of SVD:

$$\begin{aligned} \mathbf{D}\mathbf{D}^{T} &= \left(\mathbf{M}\boldsymbol{\Sigma}\mathbf{T}^{T}\right)\left(\mathbf{M}\boldsymbol{\Sigma}\mathbf{T}^{T}\right)^{T} = \left(\mathbf{M}\boldsymbol{\Sigma}\mathbf{T}^{T}\right)\left(\mathbf{T}^{T^{T}}\boldsymbol{\Sigma}^{T}\mathbf{M}^{T}\right) \\ &= \mathbf{M}\boldsymbol{\Sigma}\mathbf{T}^{T}\mathbf{T}\boldsymbol{\Sigma}^{T}\mathbf{M}^{T} = \mathbf{M}\boldsymbol{\Sigma}\boldsymbol{\Sigma}^{T}\mathbf{M}^{T} \\ \mathbf{D}^{T}\mathbf{D} &= \left(\mathbf{M}\boldsymbol{\Sigma}\mathbf{T}^{T}\right)^{T}\left(\mathbf{M}\boldsymbol{\Sigma}\mathbf{T}^{T}\right) = \left(\mathbf{T}^{T^{T}}\boldsymbol{\Sigma}^{T}\mathbf{M}^{T}\right)\left(\mathbf{M}\boldsymbol{\Sigma}\mathbf{T}^{T}\right) \\ &= \mathbf{T}\boldsymbol{\Sigma}^{T}\mathbf{M}^{T}\mathbf{M}\boldsymbol{\Sigma}\mathbf{T}^{T} = \mathbf{T}\boldsymbol{\Sigma}^{T}\boldsymbol{\Sigma}\mathbf{T}^{T}, \end{aligned} \tag{3.23}$$

which implies that $\mathbf{M}$ contains the eigenvectors of $\mathbf{D}\mathbf{D}^{T}$ while $\mathbf{T}$ contains the eigenvectors of $\mathbf{D}^{T}\mathbf{D}$. The corresponding eigenvalues are the diagonal entries of $\boldsymbol{\Sigma}^{T}\boldsymbol{\Sigma}$, or $\boldsymbol{\Sigma}^{2}$, since $\boldsymbol{\Sigma}$ is diagonal. The truncation operation is performed by ordering the eigenvalues in descending order in $\boldsymbol{\Sigma}$ and removing the smallest eigenvalue entries with their corresponding eigenvectors in $\mathbf{M}$ and $\mathbf{T}$, thereby reducing the dimension of the representation.

In MRD4SKB, we use the matrices $\mathbf{M}$ and $\mathbf{T}$ as they were denoted before. This method will be referred to as MRD4SKB_TM_LSA.

3.3.3 Probabilistic LSA (pLSA)

pLSA [64] method uses the term-document matrix to model the probability of each word-document co-occurrence as a mixture of conditionally independent multinomial distributions:

$$\begin{aligned} P(w,d) &= \sum_{c} P(z)P(d \mid z)P(w \mid z) \\ &= P(d)\sum_{z} P(z \mid d)P(w \mid z), \end{aligned} \tag{3.24}$$

where w, d, and z denote a word, a document, and a topic, respectively. Then, we use this model to generate words for documents as described below in Algorithm 2.

Algorithm 2: pLSA generative procedure.

```
for document $d$ in $D = d_1, \cdots, d_M$ do
    for word $w$ in $d = w_1, \cdots, w_N$ do
        Choose $z \sim p(z \mid d)$
        Choose $w \sim p(w \mid z)$
    end
end
```

Fig. 3.2 describes the same generative procedure in plate notation for a graphical model. In plate notation, observable variables are grayed out, whereas other variables are latent. Rectangles in the notation denote repetitions with the specified counts.

The model has a total number of $zd+wz$ parameters, which are learned through the Expectation Maximization (EM) algorithm [65] by comparing the statistics of

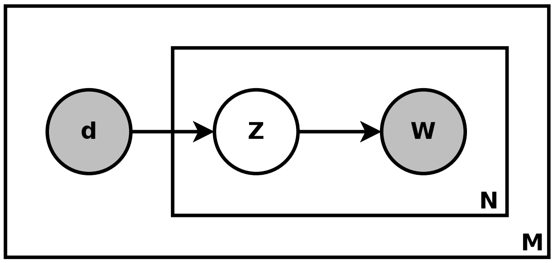

Figure 3.2: Plate notation of pLSA [2]. d denotes a document, z denotes a topic, w denotes a word, M is the number of documents, and N is the total number of words in all documents.

the generated documents with the original documents in the corpus.

In MRD4SKB, the matrix of document-topic distributions corresponds to our $\mathbf{M}$ matrix, and the matrix of topic-term vectors corresponds to our $\mathbf{T}$ matrix. This method will be denoted as MRD4SKB_TM_pLSA.

3.3.4 Latent Dirichlet Allocation (LDA)

LDA [66] is another generative statistical model used in topic modeling. The generation process is more involved than pLSA, but using the EM algorithm for parameter learning remains the same. In LDA, the probability distributions $p(z \mid d)$ and $p(w \mid z)$ are assumed to be multinomial distributions. These distributions are determined by Dirichlet distributions of parameter α and η, respectively. LDA is a generalization of pLSA, where pLSA is shown to be equivalent to LDA under a uniform Dirichlet prior distribution [67]. The generation algorithm and the plate notation for LDA generative process are given in Algorithm 3 and Fig. 3.3, respectively.

In MRD4SKB, the matrix of document-topic distributions corresponds to our $\mathbf{M}$ matrix, and the matrix of topic-term vectors corresponds to our $\mathbf{T}$ matrix. This method will be denoted as MRD4SKB_TM_LDA.

The topic modeling methods explained in this section are used in MRD4SKB

Algorithm 3: LDA generative procedure.

```
Choose $\boldsymbol{\theta}_i \sim Dir(\alpha)$ for $i \in \{1, \cdots, M\}$
Choose $\boldsymbol{\beta}_k \sim Dir(\eta)$ for $k \in \{1, \cdots, K\}$
for $word\ w_{i,j}\ in\ i \in \{1, \cdots, M\}, j \in \{1, \cdots, N\}$ do
    Choose $z_{i,j} \sim multinomial(\boldsymbol{\theta}_i)$
    Choose $w_{i,j} \sim multinomial(\boldsymbol{\beta}_{z_{i,j}})$
end
```

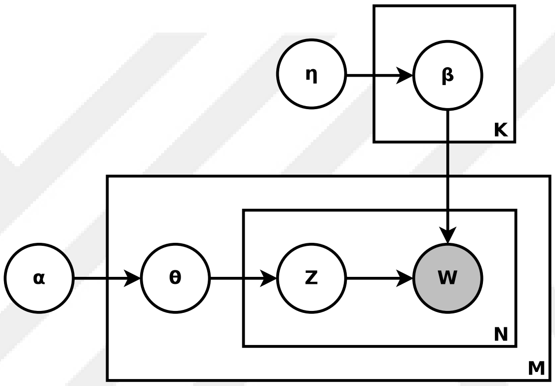

Figure 3.3: Plate notation of LDA [2]. α and η are the Dirichlet priors, θ and β are the multinomial distribution coefficients, z denotes a topic, w denotes a word, K is the number of topics, M is the number of documents, and N is the total number of words in all documents.

for decomposing the MRD term-document matrix into a term-topic and a topic-document matrix. The documents are the definitions, and the terms are the individual words within the MRD in this context. It should be noted that depending on the particular MRD, the set of all terms used in the definitions, $\mathbf{t}'$, and the set of all words defined in the dictionary, $\mathbf{t}''$, may not be identical. Ideally, every term used in a definition should be defined in a complete and consistent MRD. However, it is not practical to have such an assumption for real-world MRDs. Therefore, an intersection of these terms, $\mathbf{t}$, is used in our term-document matrices, as previously defined in Eq. 3.3.

The topic-document matrix represents the topic distribution of the documents after applying topic modeling to the term-document matrix. Each document corresponds to a definition entry from the MRD. The obtained topics for a document represent the sememes of the word defined by that particular definition entry. Determination of sememe terms from the topic vectors is accomplished as described in Eq. 3.18. This set of MRD4SKB variants will hereafter be referred to as MRD4SKB_TM.

Chapter 4

Experiments and Results[1]

In our experiments, we utilized two MRDs extracted from the definitions of WordNet [68] and the Wiktionary [49]. The MRDs are preprocessed with SpaCy [54]. In order to obtain the term-document matrices of MRD4SKB, lemmatization, and stop-word removal steps are applied to the MRDs. The compact word-sememe matrices that resulted from this procedure are regarded as SKBs. We then utilized these SKBs in specific NLP tasks to enhance their performance.

The results of the experiments will be reported in tables throughout this section. The hyperparameters used in variants of our proposed MRD4SKB_KP models are provided in square brackets in the following order:

$$MRD4SKB\_KP[mrd,\ N,\ \mu,\ k],$$

where mrd is the used MRD descriptor (wn: WordNet, wk: Wiktionary), N is the number of sememes to be determined, μ is the semantic relation factor, and k is the binary sememe annotation threshold. For the MRD4SKB_TM methods, the following notation is used:

$$MRD4SKB\_TM\_tmm[mrd,\ N],$$

[1] Source codes, data, and information to reproduce our experiments will be available at https://github.com/koc-lab/mrd2skb.

where *tmm* is the utilized topic modeling method (NMF, LSA, pLSA, LDA), and *mrd* and N denote the same parameters as given before.

We present the quantitative and qualitative results of our experiments in the following sections. Moreover, the hyperparameter analysis of the proposed methods will be presented as well.

4.1 Quantitative Results

First, as part of the intrinsic evaluation, an established consistency assessment method named Consistency Check with Sememe Annotations (CCSA) is used to evaluate the consistency of sememe annotations in SKBs generated by the proposed MRD4SKB process [69]. In CCSA, a small set of senses are subjected to the sememe prediction task. The sememe prediction is performed using the sememe annotations of other senses in the SKB, based on collaborative filtering [70]. Superior sememe prediction performance in this task indicates that semantically related senses are annotated with similar sememes. This, in turn, means that the SKB allows consistent sememe annotations, and thus is an appropriate model for representing the semantics of the natural language. In our experiments, the performance of our proposed MRD4SKB variants is compared with a set of baselines. In addition to the manually crafted SKB HowNet, two versions of the semi-automatically built EDKSBs are selected for comparison [38]. Sememe annotation consistency results of our experiments are displayed with mean average precision (MAP) and F1 metrics in Table 4.1.

Beyond the intrinsic evaluation task, four extrinsic evaluation tasks have been performed as part of our experiments. Language Modeling (LM) is the first extrinsic evaluation task. Long short-term memory (LSTM) [71] based language models enhanced with sememe knowledge are used in this task. Integration of SKB data into the base language models is achieved by a method named Sememe-Driven Language Model (SDLM) [72]. Two LSTM variants are used as base language models: Tied LSTM [73] and Averaged Stochastic Gradient Descent (ASGD)

Table 4.1: CCSA Results.

SKB	MAP ↑	F1 ↑
HowNet	0.93	**0.91**
EDSKB	0.88	0.86
EDSKB_s	**0.95**	**0.91**
MRD4SKB_KP[wn, 2000, 0.2, 0.2]	0.88	0.84
MRD4SKB_KP[wn, 2000, 0.2, 0.8]	0.86	0.85
MRD4SKB_KP[wn, 2000, 0.5, 0.8]	0.87	0.88
MRD4SKB_KP[wn, 3000, 0.5, 0.8]	0.93	0.90
MRD4SKB_KP[wk, 2000, 0.2, 0.2]	0.85	0.83
MRD4SKB_KP[wk, 2000, 0.2, 0.8]	0.83	0.82
MRD4SKB_KP[wk, 2000, 0.5, 0.8]	0.84	0.82
MRD4SKB_KP[wk, 3000, 0.5, 0.8]	0.88	0.84
MRD4SKB_TM_NMF[wn, 1000]	0.80	0.78
MRD4SKB_TM_LSA[wn, 1000]	0.82	0.80
MRD4SKB_TM_pLSA[wn, 1000]	0.83	0.81
MRD4SKB_TM_LDA[wn, 1250]	0.87	0.83

Weight-Dropped LSTM (AWD LSTM) [74]. For the evaluations, two benchmark LM datasets are utilized: the Penn Treebank (PTB) [75], and WikiText-2 [76]. Models incorporated with the original HowNet and EDSKB variants are used as baselines in addition to the vanilla models. LM task perplexity results for both validation and test sets of the benchmark datasets are provided in Table 4.2 for Tied LSTM-based models and in Table 4.3 for AWD LSTM-based models, respectively.

The second extrinsic task is Natural language inference (NLI). The SNLI dataset is utilized for this task, which consists of pairs of sentences containing a premise and a hypothesis [77]. Based on their relations, these sentence pairs are manually classified as one of the following labels: “entailment", “contradiction", or “neutral". Different RNN architecture variants were used as base models in our tests. These RNN architectures are LSTM, Gated recurrent unit (GRU) [78], and their bidirectional variants [79]. SKB information is incorporated into these

Table 4.2: SDLM Tied LSTM Perplexity (↓) Results.

Dataset	PTB		WikiText-2	
Model	Valid	Test	Valid	Test
Tied LSTM	63.92	63.98	53.10	51.41
+HowNet	58.93	58.95	48.43	47.28
+EDSKB	**58.81**	**58.82**	**43.48**	**42.15**
+EDSKB_s	60.17	60.15	45.18	42.59
+MRD4SKB_KP[wn, 2000, 0.2, 0.2]	59.23	59.23	45.31	45.47
+MRD4SKB_KP[wn, 2000, 0.2, 0.8]	59.12	59.08	45.09	45.25
+MRD4SKB_KP[wn, 2000, 0.5, 0.8]	59.28	59.32	45.12	45.58
+MRD4SKB_KP[wn, 3000, 0.5, 0.8]	58.87	58.96	44.63	44.12
+MRD4SKB_KP[wk, 2000, 0.2, 0.2]	59.35	59.45	45.62	45.49
+MRD4SKB_KP[wk, 2000, 0.2, 0.8]	59.20	59.13	45.55	45.38
+MRD4SKB_KP[wk, 2000, 0.5, 0.8]	59.40	59.45	45.34	45.71
+MRD4SKB_KP[wk, 3000, 0.5, 0.8]	59.34	59.51	44.92	44.82
+MRD4SKB_TM_NMF[wn, 1000]	59.70	59.81	45.70	45.92
+MRD4SKB_TM_LSA[wn, 1000]	59.62	59.79	45.65	45.77
+MRD4SKB_TM_pLSA[wn, 1000]	59.58	59.65	45.59	45.71
+MRD4SKB_TM_LDA[wn, 1250]	59.47	59.51	45.56	45.64

RNN models using a method called SememeCell [20]. We used this method to embed the sememe information from our automatically generated MRD4SKBs into the base models. As usual, The vanilla RNN models infused with HowNet and EDSKB data are used as baselines. Additionally, two other baselines were utilized. A Pseudo SKB is used as one baseline, created by annotating words with random labels instead of sememes. Integrating unmodified dictionary definitions directly into the RNNs is used as another baseline. The test results of SNLI are shown in Table 4.4.

The third extrinsic task is Sentiment Analysis (SA). The CR dataset was used to evaluate this task [80]. The dataset contains approximately $8,000$ product reviews. Each review is manually labeled either “positive" or “negative" based on

Table 4.3: SDLM AWD LSTM Perplexity (↓) Results.

Dataset	PTB		WikiText-2	
Model	Valid	Test	Valid	Test
AWD LSTM	58.89	59.24	45.29	44.13
+HowNet	58.95	58.92	46.84	45.29
+EDSKB	**56.94**	**57.13**	**42.44**	**41.25**
+EDSKB_s	58.63	58.59	43.85	43.95
+MRD4SKB_KP[wn, 2000, 0.2, 0.2]	58.16	58.20	43.57	43.61
+MRD4SKB_KP[wn, 2000, 0.2, 0.8]	58.04	58.17	43.49	43.58
+MRD4SKB_KP[wn, 2000, 0.5, 0.8]	57.98	58.11	43.40	43.42
+MRD4SKB_KP[wn, 3000, 0.5, 0.8]	57.84	57.99	43.15	43.31
+MRD4SKB_KP[wk, 2000, 0.2, 0.2]	58.27	58.24	43.57	43.63
+MRD4SKB_KP[wk, 2000, 0.2, 0.8]	58.23	58.22	43.61	43.64
+MRD4SKB_KP[wk, 2000, 0.5, 0.8]	58.39	58.25	43.56	43.52
+MRD4SKB_KP[wk, 3000, 0.5, 0.8]	58.22	58.27	43.44	43.47
+MRD4SKB_TM_NMF[wn, 1000]	58.73	58.67	43.82	43.90
+MRD4SKB_TM_LSA[wn, 1000]	58.65	58.62	43.76	43.88
+MRD4SKB_TM_pLSA[wn, 1000]	58.63	58.58	43.71	43.75
+MRD4SKB_TM_LDA[wn, 1250]	58.52	58.48	43.65	43.72

the conveyed emotion. The models built and used in the SNLI task are also used in the CR task. Table 4.5 provides the accuracy results for the CR dataset.

The fourth and final extrinsic task is Textual Adversarial Attacking (TAA), which helps reveal the vulnerabilities and improve the robustness of neural network models [81]. In adversarial attacks, adversarial examples are created maliciously by perturbing the original model input to fool a model [82]. Textual adversarial attacking uses word-level attacks based on word substitution in general, showing better attack performance overall [83].

We use a sememe-based word substitution strategy for TAA, which regards two words as substitutes if one sense of each word has the same sememes, according to an SKB [84]. Several baseline methods are incorporated. Using synonym-based

Table 4.4: SNLI Accuracy (↑) Results.

Model	LSTM	GRU	BiLSTM	BiGRU
vanilla	80.66	82.00	81.30	81.61
+Pseudo	81.28	80.90	81.91	82.07
+HowNet	81.87	82.90	82.55	83.15
+Definition	81.62	82.80	81.10	83.22
+EDSKB	**82.82**	**83.18**	**82.54**	**83.55**
+EDSKB_s	81.78	82.10	82.11	82.35
+MRD4SKB_KP[wn, 2000, 0.2, 0.2]	81.46	81.34	81.41	81.73
+MRD4SKB_KP[wn, 2000, 0.2, 0.8]	81.58	81.48	81.62	81.91
+MRD4SKB_KP[wn, 2000, 0.5, 0.8]	81.63	81.94	81.51	81.84
+MRD4SKB_KP[wn, 3000, 0.5, 0.8]	81.85	82.87	81.76	82.13
+MRD4SKB_KP[wk, 2000, 0.2, 0.2]	81.20	81.14	81.28	81.54
+MRD4SKB_KP[wk, 2000, 0.2, 0.8]	81.17	81.26	81.50	81.74
+MRD4SKB_KP[wk, 2000, 0.5, 0.8]	81.11	81.79	81.46	81.67
+MRD4SKB_KP[wk, 3000, 0.5, 0.8]	81.42	81.64	81.53	81.80
+MRD4SKB_TM_NMF[wn, 1000]	81.02	81.10	81.08	81.48
+MRD4SKB_TM_LSA[wn, 1000]	81.07	81.05	81.28	81.54
+MRD4SKB_TM_pLSA[wn, 1000]	81.13	81.20	81.33	81.59
+MRD4SKB_TM_LDA[wn, 1250]	81.21	81.34	81.45	81.65

word substitution utilizing WordNet, using definition-based word substitutions utilizing BERT-based word vectors, and using other sememe-based word substitutions utilizing other SKBs: HowNet, EDSKB, and EDSKB_s [38].

BiLSTM and BERT models are used as victim models [84]. Sentiment analysis with SST-2 dataset, which contains 10000 labeled sentences from movie reviews, is used as the evaluation task [85]. The effectiveness of an attack method is determined by its attack success rate (ASR), and the quality of its adversarial examples is assessed using three metrics: word modification rate (%M), grammatical error increase rate (IGE), and perplexity given by GPT-2 (PPL) [86]. Lower is better in these three metrics in terms of adversarial example quality. TAA test results are given in Table 4.6 and 4.7.

Table 4.5: CR Accuracy (↑) Results.

Model	LSTM	GRU	BiLSTM	BiGRU
Vanilla	74.17	76.37	77.62	78.76
+Pseudo	73.96	75.44	76.16	78.20
+HowNet	76.47	78.57	77.66	76.25
+Definition	76.29	78.20	77.19	77.77
+EDSKB	**77.51**	**79.68**	**78.95**	**78.88**
+EDSKB_s	75.09	77.54	76.90	78.18
+MRD4SKB_KP[wn, 2000, 0.2, 0.2]	75.88	77.41	77.47	77.91
+MRD4SKB_KP[wn, 2000, 0.2, 0.8]	76.54	78.73	77.92	78.11
+MRD4SKB_KP[wn, 2000, 0.5, 0.8]	76.25	79.01	77.84	78.17
+MRD4SKB_KP[wn, 3000, 0.5, 0.8]	77.18	77.98	78.24	78.42
+MRD4SKB_KP[wk, 2000, 0.2, 0.2]	75.64	77.31	77.39	77.82
+MRD4SKB_KP[wk, 2000, 0.2, 0.8]	76.48	78.58	77.77	77.90
+MRD4SKB_KP[wk, 2000, 0.5, 0.8]	76.02	78.61	77.84	78.07
+MRD4SKB_KP[wk, 3000, 0.5, 0.8]	76.82	77.58	78.01	77.94
+MRD4SKB_TM_NMF[wn, 1000]	75.01	77.24	77.12	77.05
+MRD4SKB_TM_LSA[wn, 1000]	75.14	77.32	77.18	77.07
+MRD4SKB_TM_pLSA[wn, 1000]	75.17	77.38	77.34	77.17
+MRD4SKB_TM_LDA[wn, 1250]	75.34	77.46	77.39	77.25

4.2 Discussion of Results

The intrinsic and extrinsic task evaluation results were presented throughout Tables 4.1 to 4.7. In each table, the best scores of manually and semi-manually crafted SKBs are emboldened, whereas the best scores of the fully automatically generated SKBs generated by the proposed MRD4SKB methods are underlined. Upward and downward arrows are placed next to the metric names to indicate whether higher or lower values are better for a particular metric in each table. Upon closer examination, proposed MRD4SKB methods are observed to perform on par with methods that require linguistic expertise and special dictionaries. The quantitative results indicate that the Kronecker product-based MRD4SKB_KP method with $3,000$ topics performs best among the MRD4SKB approaches.

Table 4.6: TAA ASR and Quality Results for BiLSTM.

Attack Method	ASR (↑)	%M (↓)	%IGE (↓)	PPL (↓)
+Synonym	79.00	10.45	7.59	593.09
+Definition	90.00	8.76	7.56	518.71
+Hownet	93.60	9.02	2.57	**468.92**
+EDSKB	26.50	**8.27**	3.77	538.46
+EDSKB_s	**94.00**	8.29	**1.27**	507.34
+MRD4SKB_KP[wn, 2000, 0.2, 0.2]	91.14	9.02	4.26	527.91
+MRD4SKB_KP[wn, 2000, 0.2, 0.8]	91.27	8.90	4.24	526.88
+MRD4SKB_KP[wn, 2000, 0.5, 0.8]	91.61	8.87	4.04	525.81
+MRD4SKB_KP[wn, 3000, 0.5, 0.8]	92.08	8.82	3.47	521.14
+MRD4SKB_KP[wk, 2000, 0.2, 0.2]	88.96	9.34	4.46	538.16
+MRD4SKB_KP[wk, 2000, 0.2, 0.8]	89.32	9.11	4.40	535.71
+MRD4SKB_KP[wk, 2000, 0.5, 0.8]	89.90	9.04	4.30	535.05
+MRD4SKB_KP[wk, 3000, 0.5, 0.8]	90.27	9.02	4.28	532.31
+MRD4SKB_TM_NMF[wn, 1000]	88.19	9.54	5.42	548.99
+MRD4SKB_TM_LSA[wn, 1000]	88.35	9.48	5.15	547.46
+MRD4SKB_TM_pLSA[wn, 1000]	88.70	9.43	4.89	545.33
+MRD4SKB_TM_LDA[wn, 1250]	90.00	9.36	4.62	544.82

MRD4SKB methods that utilized WordNet as the MRD had higher performance in general. Hence, the properties of the MRD are seen to have a noticeable effect on the quality of the auto-generated SKBs. The best-performing MRD4SKB method is observed to outperform HowNet and EDSKB_s in most extrinsic task results, except for the CCSA and TAA results. The performance of best-performing MRD4SKB methods remains slightly below that of EDSKB in quantitative tasks, but this is offset by fully automatizing the SKB generation process. The initial set of sememes in our SKBs is selected automatically without dependence on linguistic experts or a manually prepared CDV. Moreover, the proposed MRD4SKB methodology can efficiently utilize different MRDs for constructing SKBs, which was not achievable with previous approaches.

Table 4.7: TAA ASR and Quality Results for BERT.

Attack Method	ASR (↑)	%M (↓)	%IGE (↓)	PPL (↓)
+Synonym	81.30	9.22	8.00	576.82
+Definition	86.30	8.03	7.18	538.00
+Hownet	91.20	8.25	2.08	503.06
+EDSKB	29.70	8.10	3.36	**485.00**
+EDSKB_s	**93.30**	**7.66**	**1.07**	544.51
+MRD4SKB_KP[wn, 2000, 0.2, 0.2]	88.36	8.42	3.82	517.22
+MRD4SKB_KP[wn, 2000, 0.2, 0.8]	89.61	8.41	3.15	517.19
+MRD4SKB_KP[wn, 2000, 0.5, 0.8]	90.13	8.29	2.79	516.06
+MRD4SKB_KP[wn, 3000, 0.5, 0.8]	90.25	8.21	2.74	514.91
+MRD4SKB_KP[wk, 2000, 0.2, 0.2]	86.69	9.08	4.10	535.45
+MRD4SKB_KP[wk, 2000, 0.2, 0.8]	86.72	9.07	4.02	528.44
+MRD4SKB_KP[wk, 2000, 0.5, 0.8]	87.54	8.91	3.97	525.26
+MRD4SKB_KP[wk, 3000, 0.5, 0.8]	87.64	8.80	3.83	519.59
+MRD4SKB_TM_NMF[wn, 1000]	84.55	9.83	4.86	549.47
+MRD4SKB_TM_LSA[wn, 1000]	85.67	9.35	4.66	539.07
+MRD4SKB_TM_pLSA[wn, 1000]	85.78	9.25	4.54	538.46
+MRD4SKB_TM_LDA[wn, 1250]	85.89	9.13	4.31	537.55

4.3 Qualitative Examples

Sample qualitative examples are provided from our autogenerated SKBs. Two examples of sememe annotations with the proposed MRD4SKB and other baseline methods are presented in Table 4.8.

The word `hospital` is the first qualitative example. The Oxford Dictionary defines this word as "a large building where people who are ill or injured are given medical treatment and care". This word has only one sense. Examining the sememe annotations in the table reveals that while HowNet can express the general meaning of a word, more specific sememe entries are produced by dictionary-based methods. Sememe entries like `health` and `surgical` that were annotated by the MRD4SKB_KP are examples of more specific entries. The word `tweet` is the second qualitative example. The Oxford Dictionary has two senses for this word. The first sense is defined as "the short, high sound made by a small bird".

Table 4.8: Qualitative Results.

Word	SKB	Sememes
hospital	Hownet	InstitutePlace, medical, doctor, disease
	EDSKB	medical, large, injure, receive, people, treatment, build, sick
	MRD4SKB_KP	health, receive, treatment, care, given, institution, medical, people, surgical
	MRD4SKB_TM_NMF	treatment, given, people
	MRD4SKB_TM_LSA	treatment, given, people
	MRD4SKB_TM_pLSA	receive, treatment, given, institution, surgical
	MRD4SKB_TM_LDA	health, receive, treatment, given, institution
tweet	Hownet	InstitutePlace, ProperName, produce, software, LookFor, document, information, internet
	EDSKB	**Sense 1:** bird, make, high, small, short, sound **Sense 2:** service, message, network, short, send, use, social
	MRD4SKB_KP	**Sense 1:** bird, small, sound, weak **Sense 2:** computer, message, popular, short, social, text
	MRD4SKB_TM_NMF	**Sense 1:** bird, small, sound **Sense 2:** short, social, site
	MRD4SKB_TM_LSA	**Sense 1:** small, sound **Sense 2:** short, social
	MRD4SKB_TM_pLSA	**Sense 1:** small, sound **Sense 2:** popular, short, social
	MRD4SKB_TM_LDA	**Sense 1:** small, sound **Sense 2:** popular, short, social, site

The second sense is defined as “a message sent using the Twitter social media service”, a more modern-time definition than the first. It is observed that HowNet successfully covers the second sense but falls short of covering the first sense. As in this example, more word senses can be expressed in a dictionary-based SKB, depending on the utilized MRD. The number of generated sememes per word is generally lower for topic modeling-based MRD4SKB approaches. This is partly expected, as the number of topics was lower in the MRD4SKB_TM methods, compared to MRD4SKB_KP methods. Using more topics in the topic modeling-based methods were impractical due to convergence issues encountered during implementation. Therefore, the number of sememes was limited to around $1,500$ in the MRD4SKB_TM methods. The number of sememes in the MRD4SKB_KP method could reach around $3,000$ in comparison. It should be noted that MRDs contain relatively small documents. Hence, alternative short-text topic modeling approaches can be considered for future research.

4.4 Analysis of Hyperparameters

In order to obtain the best performance on intrinsic and extrinsic tasks using our MRD4SKB methods, hyperparameter fine-tuning is performed. The task of NLI is chosen for the hyperparameter optimization. The SNLI dataset and the LSTM base model are selected for this purpose. Then, keeping the rest of the experimental setup fixed, a single parameter of the MRD4SKB model is altered. The effect of these changes on the task performance metrics is then analyzed. The performance metric becomes the SNLI accuracy for the selected optimization task.

First, the effect of the number of sememes in the initial sememe set, N, on the task performance is examined. Results of this examination are given in Fig. 4.1. The effect of the number of sememes on task performance is considerably high, as there is a wide variation in performance for different values of N. The total number of distinct sememes in the SKB can be considered as the dimensionality of the semantic space represented by the SKB. Since each term comprises a combination of these sememes in the SKB, the initial sememe set essentially acts as a basis for this semantic space.

Low N values create dense representations, where components of the semantic space get highly coupled with each other. On the other end of the spectrum, overly sparse representations are generated using high values of N. In this case, very few elements are contained within distinct components of the semantic space, and the semantic connections between the terms can not be captured. The best performing N value for MRD4SKB was $3,000$ among the searched values. This number remarkably matches the number of sememes in the manually constructed HowNet, reassuring the validity of our method. HowNet utilizes the most commonly used Chinese words as its initial sememe set, and our experiments indicate that the number of commonly used Chinese words is close to optimal for the number of sememes in an SKB.

Next, the effect of the semantic relation factor, μ, on task performance is examined. Results are shown in Fig. 4.2. It is observed that a semantic relation

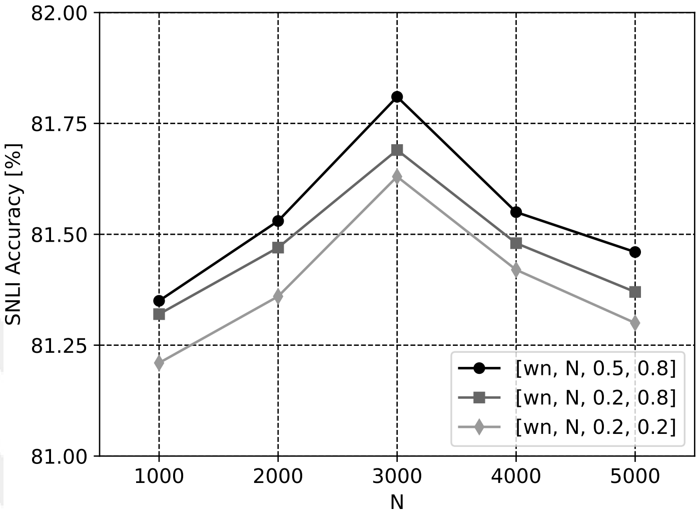

Figure 4.1: Accuracy results of $MRD4SKB\_KP$ models with varying number of total sememes, N, on the test set of SNLI.

factor of $\mu = 0.5$ works well for our application.

The effect of the binary sememe annotation threshold, k, on the performance is also examined. Results are displayed in Fig. 4.3. In the extreme case when $k = 1$, only the sememes directly used in the definition of a term can be used in the sememe annotation of that term, and no indirect sememes can be used. The results demonstrate that semantic compositionality should be utilized in moderation during the generation of the SKB to maximize the performance on extrinsic tasks.

Finally, the effect of the number of sememes, N, on the performance of topic modeling-based methods is examined. In the topic modeling-based methods, the number of sememes is equivalent to the number of topics. Results are reported in Fig. 4.4. Using higher numbers of topics in topic modeling-based approaches frequently caused convergence issues, and most of these methods had the best performance at around $1,000$ topics. In general, the performance of topic modeling-based methods remained lower than that of the Kronecker product-based method on the hyperparameter evaluation task.

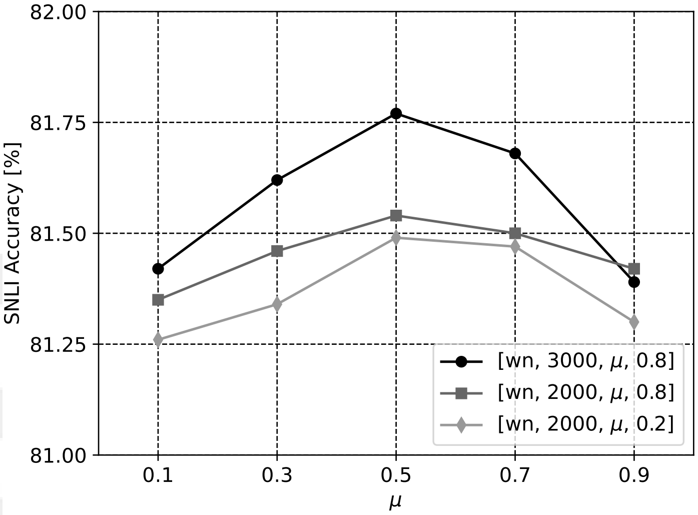

Figure 4.2: Accuracy results of $MRD4SKB\_KP$ models with varying semantic relation factor, μ, on the test set of SNLI.

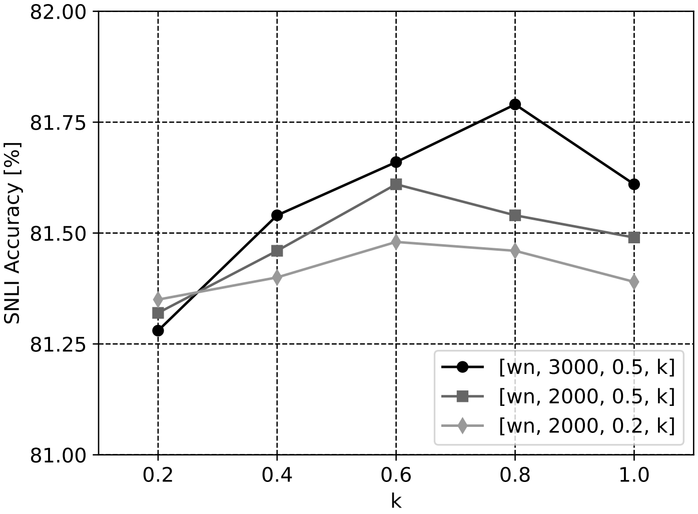

Figure 4.3: Accuracy results of $MRD4SKB\_KP$ models with varying binary sememe annotation threshold, k, on the test set of SNLI.

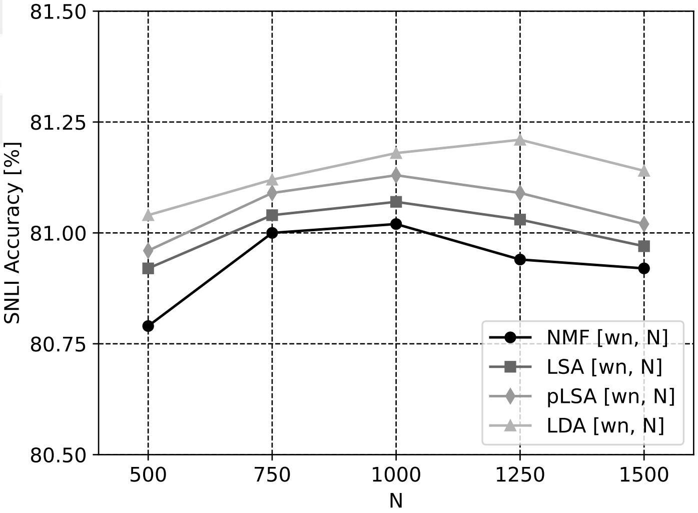

Figure 4.4: Accuracy results of $MRD4SKB\_TM\_tmm$ models with varying number of total sememes, N, on the test set of SNLI. Note that this corresponds to the number of topics for the topic modeling-based methods.

Chapter 5

Conclusions

We proposed MRD4SKB, a fully computational and automatic framework for constructing SKBs from arbitrary MRDs. Matrix factorization and topic modeling-based alternatives of our proposed MRD4SKB methodology are presented. Through intrinsic and extrinsic evaluation tasks and using two different English language MRDs, the validity of the SKBs automatically generated with our MRD4SKB methods are experimentally demonstrated. The performance of the constructed SKBs on various NLP tasks is demonstrated. The effects of the specified hyperparameters are individually analyzed. Quantitative examples show what to expect from SKBs constructed with our proposed methods.

Our framework could automatically produce SKBs, that outperform the manually created HowNet and are on par with other baseline SKBs that rely on special manually prepared CDVs. Moreover, our automatized approach is generic and can be applied to different MRDs without requiring a specially crafted dictionary.

Construction of SKBs from readily available MRDs through a fully computational and automatic framework can unlock directions of research that were not previously possible. Furthermore, such a framework can expedite existing research on improving a wide range of high-level NLP tasks by incorporating sememe knowledge.

Bibliography

[1] F. Qi, Y. Lin, M. Sun, H. Zhu, R. Xie, and Z. Liu, "Cross-lingual lexical sememe prediction," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (Stroudsburg, PA, USA), Association for Computational Linguistics, 2018.

[2] B. V. Barde and A. M. Bainwad, "An overview of topic modeling methods and tools," in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, pp. 745–750, IEEE, 2017.

[3] L. Bloomfield, "A set of postulates for the science of language," *Language*, vol. 2, no. 3, pp. 153–164, 1926.

[4] Z. Dong and Q. Dong, "HowNet - a hybrid language and knowledge resource," in *International Conference on Natural Language Processing and Knowledge Engineering, 2003. Proceedings. 2003*, IEEE, 2004.

[5] F. Qi, R. Xie, Y. Zang, Z. Liu, and M. Sun, "Sememe knowledge computation: a review of recent advances in application and expansion of sememe knowledge bases," *Frontiers of Computer Science*, vol. 15, no. 5, pp. 1–11, 2021.

[6] Y. Liu, M. Zhang, and D. Ji, "End to end chinese lexical fusion recognition with sememe knowledge," in *Proceedings of the 28th International Conference on Computational Linguistics*, (Stroudsburg, PA, USA), International Committee on Computational Linguistics, 2020.

[7] Y. Niu, R. Xie, Z. Liu, and M. Sun, "Improved word representation learning with sememes," in *Proceedings of the 55th Annual Meeting of the Association*

for Computational Linguistics (Volume 1: Long Papers), (Stroudsburg, PA, USA), Association for Computational Linguistics, 2017.

[8] F. Xianghua, L. Guo, G. Yanyan, and W. Zhiqiang, "Multi-aspect sentiment analysis for chinese online social reviews based on topic modeling and HowNet lexicon," *Knowl. Based Syst.*, vol. 37, pp. 186–195, 2013.

[9] L. Yang, C. Kong, Y. Chen, Y. Liu, Q. Fan, and E. Yang, "Incorporating sememes into chinese definition modeling," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 1669–1677, 2020.

[10] J. Li, Y. Bao, S. Huang, X. Dai, and J. Chen, "Explicit semantic decomposition for definition generation," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 708–717, 2020.

[11] J. Qiang, X. Lu, Y. Li, Y.-H. Yuan, and X. Wu, "Chinese lexical simplification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021.

[12] X. Zeng, C. Yang, C. Tu, Z. Liu, and M. Sun, "Chinese liwc lexicon expansion via hierarchical classification of word embeddings with sememe attention," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[13] D. Cai, J. Sun, G. Zhang, D. Lv, Y. Dong, Y. Song, and C. Yu, "Hownet based chinese question classification," in *Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation*, pp. 366–369, 2006.

[14] Y. Gu, J. Yan, H. Zhu, Z. Liu, R. Xie, M. Sun, F. Lin, and L. Lin, "Language modeling with sparse product of sememe experts," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (Stroudsburg, PA, USA), Association for Computational Linguistics, 2018.

[15] Z. Wen, L. Gui, Q. Wang, M. Guo, X. Yu, J. Du, and R. Xu, "Sememe knowledge and auxiliary information enhanced approach for sarcasm detection," *Information Processing & Management*, vol. 59, no. 3, p. 102883, 2022.

[16] T. Young, E. Cambria, I. Chaturvedi, H. Zhou, S. Biswas, and M. Huang, "Augmenting end-to-end dialogue systems with commonsense knowledge," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[17] S. Ahn, H. Choi, T. Pärnamaa, and Y. Bengio, "A neural knowledge language model," *arXiv preprint arXiv:1608.00318*, 2016.

[18] B. Yang and T. Mitchell, "Leveraging knowledge bases in LSTMs for improving machine reading," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Stroudsburg, PA, USA), Association for Computational Linguistics, 2017.

[19] P. Parthasarathi and J. Pineau, "Extending neural generative conversational model using external knowledge sources," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, (Stroudsburg, PA, USA), Association for Computational Linguistics, 2018.

[20] Y. Qin, F. Qi, S. Ouyang, Z. Liu, C. Yang, Y. Wang, Q. Liu, and M. Sun, "Improving sequence modeling ability of recurrent neural networks via sememes," *IEEE ACM Trans. Audio Speech Lang. Process.*, vol. 28, pp. 2364–2373, 2020.

[21] Y. Zhang, C. Yang, Z. Zhou, and Z. Liu, "Enhancing transformer with sememe knowledge," in *Proceedings of the 5th Workshop on Representation Learning for NLP*, (Stroudsburg, PA, USA), Association for Computational Linguistics, 2020.

[22] F. Qi, C. Yang, Z. Liu, Q. Dong, M. Sun, and Z. Dong, "OpenHowNet: An open sememe-based lexical knowledge base," *arXiv preprint arXiv:1901.09957*, 2019.

[23] J. Novikova, O. Lemon, and V. Rieser, "Crowd-sourcing nlg data: Pictures elicit better data," *arXiv preprint arXiv:1608.00339*, 2016.

[24] Y. Mrabet, P. Vougiouklis, H. Kilicoglu, C. Gardent, D. Demner-Fushman, J. Hare, and E. Simperl, "Aligning texts and knowledge bases with semantic sentence simplification," in *Proceedings of the 2nd International Workshop on Natural Language Generation and the Semantic Web (WebNLG 2016)*, pp. 29–36, 2016.

[25] D. Gildea and D. Jurafsky, "Automatic labeling of semantic roles," *Computational linguistics*, vol. 28, no. 3, pp. 245–288, 2002.

[26] G. Lampouras and I. Androutsopoulos, "Extracting linguistic resources from the web for concept-to-text generation," *arXiv preprint arXiv:1810.13414*, 2018.

[27] R. Lebret, D. Grangier, and M. Auli, "Neural text generation from structured data with application to the biography domain," *arXiv preprint arXiv:1603.07771*, 2016.

[28] E. Manishina, B. Jabaian, S. Huet, and F. Lefevre, "Automatic corpus extension for data-driven natural language generation," in *10th International Conference on Language Resources and Evaluation (LREC)*, pp. 3624–3631, 2016.

[29] R. Xie, X. Yuan, Z. Liu, and M. Sun, "Lexical sememe prediction via word embeddings and matrix factorization.," in *IJCAI*, pp. 4200–4206, 2017.

[30] W. Li, X. Ren, D. Dai, Y. Wu, H. Wang, and X. Sun, "Sememe prediction: Learning semantic knowledge from unstructured textual wiki descriptions," *arXiv preprint arXiv:1808.05437*, 2018.

[31] M. Bai, P. Lv, and X. Long, "Lexical sememe prediction with rnn and modern chinese dictionary," in *2018 14th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pp. 825–830, IEEE, 2018.

[32] H. Jin, H. Zhu, Z. Liu, R. Xie, M. Sun, F. Lin, and L. Lin, "Incorporating chinese characters of words for lexical sememe prediction," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, (Stroudsburg, PA, USA), Association for Computational Linguistics, 2018.

[33] F. Qi, L. Chang, M. Sun, S. Ouyang, and Z. Liu, "Towards building a multilingual sememe knowledge base: Predicting sememes for BabelNet synsets," *Proc. Conf. AAAI Artif. Intell.*, vol. 34, no. 05, pp. 8624–8631, 2020.

[34] J. Du, F. Qi, M. Sun, and Z. Liu, “Lexical sememe prediction using dictionary definitions by capturing local semantic correspondence,” *arXiv preprint arXiv:2001.05954*, 2020.

[35] B. Lyu, L. Chen, and K. Yu, “Glyph enhanced chinese character pre-training for lexical sememe prediction,” in *Findings of the Association for Computational Linguistics: EMNLP 2021*, pp. 4549–4555, 2021.

[36] F. Qi, C. Lv, Z. Liu, X. Meng, M. Sun, and H.-T. Zheng, “Sememe prediction for babelnet synsets using multilingual and multimodal information,” *arXiv preprint arXiv:2203.07426*, 2022.

[37] Y. Ye, F. Qi, Z. Liu, and M. Sun, “Going “deeper”: Structured sememe prediction via transformer with tree attention,” in *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 128–138, 2022.

[38] F. Qi, Y. Chen, F. Wang, Z. Liu, X. Chen, and M. Sun, “Automatic construction of sememe knowledge bases via dictionaries,” in *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pp. 4673–4686, 2021.

[39] Q. Zhou, G. Yue, and Y. Meng, “Incorporating HowNet-based semantic relatedness into chinese word sense disambiguation,” in *Lecture Notes in Computer Science*, pp. 359–370, Cham: Springer International Publishing, 2020.

[40] K. W. Gan and P. W. Wong, “Annotating information structures in chinese texts using HowNet,” in *Proceedings of the second workshop on Chinese language processing held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics* -, (Morristown, NJ, USA), Association for Computational Linguistics, 2000.

[41] L. Luo, X. Ao, Y. Song, J. Li, X. Yang, Q. He, and D. Yu, “Unsupervised neural aspect extraction with sememes,” in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*, (California), International Joint Conferences on Artificial Intelligence Organization, 2019.

[42] F. Qi, J. Huang, C. Yang, Z. Liu, X. Chen, Q. Liu, and M. Sun, "Modeling semantic compositionality with sememe knowledge," *arXiv preprint arXiv:1907.04744*, 2019.

[43] L. Zheng, F. Qi, Z. Liu, Y. Wang, Q. Liu, and M. Sun, "Multi-channel reverse dictionary model," *Proc. Conf. AAAI Artif. Intell.*, vol. 34, no. 01, pp. 312–319, 2020.

[44] H. Nie, J. Zhou, H. Wang, and M. Li, "Word similarity computing based on HowNet and synonymy thesaurus," in *Advances in Intelligent Systems and Computing*, pp. 292–305, Cham: Springer International Publishing, 2020.

[45] S. M. Lamb, "The sememic approach to structural semantics 1," *American anthropologist*, vol. 66, no. 3, pp. 57–78, 1964.

[46] S. Kubler, R. McDonald, and J. Nivre, "Dependency parsing," *Synthesis lectures on human language technologies*, vol. 1, no. 1, pp. 1–127, 2009.

[47] D. E. Walker and R. A. Amsler, "The use of machine-readable dictionaries in sublanguage analysis," *Analyzing language in restricted domains: Sublanguage description and processing*, pp. 69–83, 1986.

[48] G. A. Miller, R. Beckwith, C. Fellbaum, D. Gross, and K. J. Miller, "Introduction to WordNet: An on-line lexical database," *Int. j. lexicogr.*, vol. 3, no. 4, pp. 235–244, 1990.

[49] "Wiktionary, the free dictionary." https://en.wiktionary.org/.

[50] D. J. Velasco, A. Alba, T. G. Pelagio, B. A. Ramirez, J. C. B. Cruz, and C. Cheng, "Automatic wordnet construction using word sense induction through sentence embeddings," *arXiv preprint arXiv:2204.03251*, 2022.

[51] G. Chen and J. Su, "Towards non-ambiguous reverse dictionary," in *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pp. 1113–1120, IEEE, 2021.

[52] W. Yu, C. Zhu, Y. Fang, D. Yu, S. Wang, Y. Xu, M. Zeng, and M. Jiang, "Dict-bert: Enhancing language model pre-training with dictionary," *arXiv preprint arXiv:2110.06490*, 2021.

[53] M. Lesk, "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone," in *Proceedings of the 5th annual international conference on Systems documentation*, pp. 24–26, 1986.

[54] M. Honnibal, I. Montani, S. Van Landeghem, and A. Boyd, "spaCy: Industrial-strength Natural Language Processing in Python," 2020.

[55] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, p. 20150202, 2016.

[56] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.

[57] I. Vayansky and S. A. Kumar, "A review of topic modeling methods," *Information Systems*, vol. 94, p. 101582, 2020.

[58] D. Hall, D. Jurafsky, and C. D. Manning, "Studying the history of ideas using topic models," in *Proceedings of the 2008 conference on empirical methods in natural language processing*, pp. 363–371, 2008.

[59] G. Salton, "Some research problems in automatic information retrieval," in *ACM SIGIR Forum*, vol. 17, pp. 252–263, ACM New York, NY, USA, 1983.

[60] T. A. Rana, Y.-N. Cheah, and S. Letchmunan, "Topic modeling in sentiment analysis: A systematic review.," *Journal of ICT Research & Applications*, vol. 10, no. 1, 2016.

[61] Z. Wu, L. Lei, G. Li, H. Huang, C. Zheng, E. Chen, and G. Xu, "A topic modeling based approach to novel document automatic summarization," *Expert Systems with Applications*, vol. 84, pp. 12–23, 2017.

[62] W. Sriurai, "Improving text categorization by using a topic model," *Advanced Computing*, vol. 2, no. 6, p. 21, 2011.

[63] D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *Advances in neural information processing systems*, vol. 13, 2000.

[64] T. Hofmann, “Probabilistic latent semantic indexing,” in *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 50–57, 1999.

[65] A. P. Dempster, N. M. Laird, and D. B. Rubin, “Maximum likelihood from incomplete data via the em algorithm,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 39, no. 1, pp. 1–22, 1977.

[66] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

[67] M. Girolami and A. Kabán, “On an equivalence between plsi and lda,” in *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pp. 433–434, 2003.

[68] G. A. Miller, *WordNet: An electronic lexical database.* MIT press, 1998.

[69] Y. Liu, F. Qi, Z. Liu, and M. Sun, “Research on consistency check of sememe annotations in hownet,” *Journal of Chinese Information Processing*, 2020.

[70] R. Xie, X. Yuan, Z. Liu, and M. Sun, “Lexical sememe prediction via word embeddings and matrix factorization,” in *Proceedings of IJCAI*, 2017.

[71] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[72] Y. Gu, J. Yan, H. Zhu, Z. Liu, R. Xie, M. Sun, F. Lin, and L. Lin, “Language modeling with sparse product of sememe experts,” in *EMNLP*, 2018.

[73] W. Zaremba, I. Sutskever, and O. Vinyals, “Recurrent neural network regularization,” *arXiv preprint arXiv:1409.2329*, 2014.

[74] S. Merity, N. S. Keskar, and R. Socher, “Regularizing and optimizing lstm language models,” *arXiv preprint arXiv:1708.02182*, 2017.

[75] M. Marcus, B. Santorini, and M. A. Marcinkiewicz, “Building a large annotated corpus of english: The penn treebank,” *repository.upenn.edu*, 1993.

[76] S. Merity, C. Xiong, J. Bradbury, and R. Socher, "Pointer sentinel mixture models," *arXiv preprint arXiv:1609.07843*, 2016.

[77] S. R. Bowman, G. Angeli, C. Potts, and C. D. Manning, "A large annotated corpus for learning natural language inference," *arXiv preprint arXiv:1508.05326*, 2015.

[78] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (Doha, Qatar), pp. 1724–1734, Association for Computational Linguistics, Oct. 2014.

[79] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[80] M. Hu and B. Liu, "Mining and summarizing customer reviews," in *Proceedings of KDD*, 2004.

[81] H. Xu, Y. Ma, H.-C. Liu, D. Deb, H. Liu, J.-L. Tang, and A. K. Jain, "Adversarial attacks and defenses in images, graphs and text: A review," *International Journal of Automation and Computing*, vol. 17, pp. 151–178, 2020.

[82] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *arXiv preprint arXiv:1312.6199*, 2013.

[83] X. Wang, J. Hao, Y. Yang, and K. He, "Natural language adversarial defense through synonym encoding," in *Uncertainty in Artificial Intelligence*, pp. 823–833, PMLR, 2021.

[84] Y. Zang, F. Qi, C. Yang, Z. Liu, M. Zhang, Q. Liu, and M. Sun, "Word-level textual adversarial attacking as combinatorial optimization," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, (Online), pp. 6066–6080, Association for Computational Linguistics, July 2020.

[85] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts, “Recursive deep models for semantic compositionality over a sentiment treebank,” in *Proceedings of the 2013 conference on empirical methods in natural language processing*, pp. 1631–1642, 2013.

[86] A. Radford, J. Wu, R. Child, D. Luan, D. Amodei, and I. Sutskever, “Language models are unsupervised multitask learners,” *OpenAI Blog*, vol. 1, no. 8, 2019.