Knowledge graph embedding models for automatic commonsense knowledge acquisition

Ikhlas Mohammad Suliman Alhussien

2019

Ikhlas Mohammad Suliman Alhussien. (2019). Knowledge graph embedding models for automatic commonsense knowledge acquisition. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/102652>

<https://doi.org/10.32657/10220/47795>

**KNOWLEDGE GRAPH EMBEDDING MODELS
FOR AUTOMATIC COMMONSENSE
KNOWLEDGE ACQUISITION**

IKHLAS MOHAMMAD SULIMAN ALHUSSIEN

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

2019

KNOWLEDGE GRAPH EMBEDDING MODELS FOR AUTOMATIC COMMONSENSE KNOWLEDGE ACQUISITION

IKHLAS MOHAMMAD SULIMAN ALHUSSIEN

School of Computer Science and Engineering

A thesis submitted to the Nanyang Technological University
in partial fulfilment of the requirements for the degree of
Master of Engineering

2019

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research, is free of plagiarised materials, and has not been submitted for a higher degree to any other University or Institution.

15th February, 2019

.....
Date

.....
Ikhlas Mohammad Suliman Alhussien

Supervisor Declaration Statement

I have reviewed the content and presentation style of this thesis and declare it is free of plagiarism and of sufficient grammatical clarity to be examined. To the best of my knowledge, the research and writing are those of the candidate except as acknowledged in the Author Attribution Statement. I confirm that the investigations were conducted in accord with the ethics policies and integrity standards of Nanyang Technological University and that the research data are presented honestly and without prejudice.

15 Feb. 19

.....
Date

.....
Erik Cambria

Authorship Attribution Statement

This thesis contains material from [1] paper(s) published in the following workshop where I was the first author.

Chapter 4, section 4.1 and partially Chapter 5 are published as *Alhussien, I., Cambria, E., and NengSheng, Z. (2018). Semantically Enhanced Models for Commonsense Knowledge Acquisition. Data Mining Workshops (ICDMW), 2018 IEEE International Conference on. IEEE, 2018.*

The contributions of the co-authors are as follows:

- Asst/Prof Erik Cambria and Dr. Zhang NengSheng helped in deciding project direction as well as providing guidance and insights through out the project.
- I performed experimental design, experiments conduction, and results analysis and interpretation.
- I prepared the manuscript drafts. The manuscript was revised by Asst/Prof Erik Cambria and Dr. Zhang NengSheng.
- Asst/Prof Erik Cambria gave the final approval of the version to be submitted to the workshop.

15th February, 2019

.....
Date

.....
Ikhlas Mohammad Suliman Alhussien

Acknowledgements

“...and say: My Lord! Increase me in knowledge”

Quran, Taha, Verse No:114

First and foremost, I thank Allah, The Most Beneficent, The Most Merciful, for giving me the strength and patience to learn and work continually and complete this work.

I would like to express my sincere gratitude to my advisor Prof. Erik Cambria for helping me in developing the necessary research skills, and for encouraging me to learn and explore different areas of research. I also would like to thank my co-advisor Dr. Zhang NengSheng for his invaluable guidance and suggestions. Thanks both for your continuous supervision through my master work and research.

I would like to thank my lab mates and colleagues from our department for offering their precious help when needed.

I owe a lot to my friends who helped me stay strong in the toughest times of all. A special thank you goes to Noor for her contentious encouragement, concern, and prayers along the whole Masters journey. Israa, thank you for your unconditional support, listening, offering me advice, and for the good laugh.

I thank all my friends whom I met here at NTU especially Ahmed, and Shah. Indeed, my Master's journey would not be the same without having such an awesome company.

Last but not least, I would like to express my deepest gratitude to my parents and my siblings for being my backbone in life, I will never be able to thank you enough!

Ikhlas Alhussien

Nanyang Technological University

Aug 24, 2018

Contents

Acknowledgements	iv
Abstract	viii
List of Tables	ix
List of Figures	xi
1 Introduction	1
1.1 Motivation	1
1.2 Contributions	4
1.3 Challenges	5
1.4 Scope of Research	6
1.5 Thesis Outline	7
2 Related Work	8
2.1 Background	8
2.1.1 Commonsense knowledge	8
2.1.2 Commonsense Knowledge Bases	9
2.1.3 Knowledge Graph Embedding	13
2.1.4 Semantic Distributional Models	16
2.2 Building Commonsense Knowledge Bases	18
2.2.1 Manual Acquisition	19
2.2.2 Mining-Based Acquisition	24
2.2.3 Reasoning Based Acquisition	29
2.3 Comparison to prior work and its limitations	31
2.4 Applications	34

3	Models	36
3.1	Semantically Enhanced KGE Models for CSKA	36
3.1.1	Problem Formulation	38
3.1.2	Proposed Method	39
3.1.3	Knowledge Representation Model	40
3.1.4	Semantic Representation Model	41
3.2	Sense Disambiguated KGE Models for CSKA	45
3.2.1	Problem Formulation	47
3.2.2	Proposed Model	48
3.2.3	Sentence Embedding	48
3.2.4	Context Clustering and Sense Induction	48
3.2.5	Sense-specific Semantic embeddings	51
3.2.6	Sense-Disambiguated knowledge graph embeddings	52
4	Datasets and Experimental Setup	53
4.1	Semantically Enhanced KGE Models for CSKA	53
4.1.1	Commonsense Knowledge Graph	53
4.1.2	Semantics Embeddings	54
4.1.3	AffectiveSpace	56
4.1.4	Common Knowledge	56
4.2	Sense Disambiguated KGE Models for CSKA	60
4.2.1	Dataset and Experimental Setup	60
4.2.2	Context Clustering	62
4.2.3	Sense Embeddings	63
5	Evaluation and Discussion	65
5.1	Training	65
5.2	Experiments and Results	66
5.2.1	Knowledge base Completion	66
5.2.2	Triple Classification	73
6	Conclusion	78
6.1	Conclusion	78
6.2	Future Work	78

7	Appendix A	79
7.1	List Of publications	79
8	Appendix B	80
8.1	Abbreviation	80
	Bibliography	81

Abstract

Intelligent systems are expected to make smart human-like decisions based on accumulated commonsense knowledge of an average individual. These systems need, therefore, to acquire an understanding about uses of objects, their properties, parts and materials, preconditions and effects of actions, and many other forms of rather implicit shared knowledge. Formalizing and collecting commonsense knowledge has, thus, been an long-standing challenge for artificial intelligence research community. The availability of massive amounts of multimodal data in the web accompanied with the advancement of information extraction and machine learning together with the increase in computational power made the automation of commonsense knowledge acquisition more feasible than ever.

Reasoning models perform automatic knowledge acquisition by making rough guesses of valid assertions based on analogical similarities. A recent successful family of reasoning models termed knowledge graph embedding convert knowledge graph entities and relations into compact k-dimensional vectors that encode their global and local structural and semantic information. These models have shown outstanding performance on predicting factual assertions in encyclopedic knowledge bases, however, in their current form, they are unable to deal commonsense knowledge acquisition. Unlike encyclopedic knowledge, commonsense knowledge is concerned with abstract concepts which can have multiple meanings, can be expressed in various forms, and can be dropped from textual communication. Therefore, knowledge graph embedding models fall short of encoding the structural and semantic information associated with these concepts and subsequently, under-perform in commonsense knowledge acquisition task.

The goal of this research is to investigate semantically enhanced knowledge graph embedding models tailored to deal with the special challenges imposed by commonsense knowledge. The research presented in this report draws on the idea that providing knowledge graph embedding models with salient and focused semantic context of concepts and relations would result in enhanced vectors representations that can be effective for automatically enriching commonsense knowledge bases with new assertions.

List of Tables

2.1	Commonsense Knowledge Bases Statistics	9
2.2	Positioning the dissertation against related work. K.type : Knowledge type [CS : Commonsense; F : Factual]; K.Src : Knowledge Source [Impl. : Implicit; Expl. : Explicit]; Cov. : Coverage; Eff. : Efficiency; Prec. : Precision; Scal. : Scalability; Extr.K : Use of External Knowledge; Ambiguity : Resolve Ambiguity. . . .	33
4.1	CN30K dataset statistics	54
4.2	CN30K relation distribution statistics	55
4.3	ProBase concepts standardized by CoreNLP tool	58
4.4	Examples of CN30K matches in ProBase instances	59
4.5	Statistics of datasets for sense disambiguation model. 1-gram=number of 1-gram concepts, 2-gram= number of 2-gram concepts, etc.	60
4.6	Full datasets relations statistics	61
4.7	Count of sense-disambiguated concepts generated by different clustering thresholds	63
4.8	Cluster Inner Distance for CN_Freq5 and CN_Freq5 datasets . .	63
5.1	Concept prediction evaluation results	68
5.2	Relation prediction evaluation results	68
5.3	Concept prediction evaluation with different clustering algorithms, Dataset= CN_Freq5	71
5.4	Concept prediction evaluation with different clustering methods, Dataset= CN_Freq10	72
5.5	Relation prediction evaluation with different clustering algorithms, Dataset= CN_Freq5	72

5.6	Relation prediction evaluation with different clustering algorithms, Dataset= CN_Freq10	74
5.7	Concept Prediction with semantic vectors, Dataset=CN_Freq5, MR=Mean Rank, H@10=Hits@10	74
5.8	Concept Prediction with semantic vectors, Dataset= CN_Freq10	74
5.11	Triple classification accuracy for CN30K	75
5.9	Relation Prediction with semantic vectors, Dataset= CN_Freq5 .	77
5.10	Relation Prediction with semantic vectors, Dataset= CN_Freq10	77
5.12	Triple classification Accuracy on CN_Freq5	77
5.13	Triple classification Accuracyz on CN_Freq10	77

List of Figures

2.1	Snapshot of ConceptNet semantic network (Source: (Lenat, 1995))	12
2.2	Hourglass of Emotions (Source:(Cambria et al., 2012a))	24
3.1	Model Architecture	39
3.2	Snapshot of a knowledge graph	46
3.3	Simple illustrations of TransE and TransR (Figures adopted from (Wang et al., 2017))	52

Chapter 1

Introduction

1.1 Motivation

When we interact, our actions are based on a layer of assumptions that are assumed to be possessed by everyone and which we collectively call commonsense knowledge (CSK). This includes properties of objects, their usage and parts, emotions, motives, preconditions and effects of actions, etc. These shared assumptions are dropped from our communication in favour of faster, smarter, and more efficient interactions. Thus, our communication is narrowed to the required information necessary to define an interaction. For example, if someone asked you to “make a cup of coffee”, it is axiomatic for you to use water and coffee powder to make the coffee, hence, this knowledge is not conveyed to you explicitly. However, for a household robot to perform the same task, the mere “make a cup of coffee” does not carry enough information to define task parts; rather, the robot needs the same background knowledge that you would use in the same situation.

The ultimate goal of artificial intelligence (AI) is to build systems that can approximate human behaviour and human decision-making. Therefore, AI researchers aim to develop machines that can approximate human level in solving problems and achieving goals. It is, therefore, a pre-request to provide these machines with the commonsense knowledge that humans possess in a machine readable format, in addition to reasoning tools to perform inference over the knowledge. Towards this endeavour, AI researchers have invested massive efforts to recall commonsense knowledge hidden in their mind and codify it into

knowledge bases (KBs). However, these efforts have faced challenges related to the characteristics of commonsense knowledge such as being implicit, easy to identify but hard to recall, culture and context dependent, etc. During early stages of commonsense knowledge acquisition (CSKA), AI researchers have thus relied on manual annotation by system experts to formalize and codify valid assertions as in Cyc (Lenat, 1995), SUMO ontology (Niles and Pease, 2001), HowNet (Zhendong and Qiang, 2006), and Open Mind Common Sense (OMCS) (Singh et al., 2002). To increase the efficiency of manual knowledge gathering, researchers have then resorted to collective efforts through public platforms such as crowd-sourcing websites and games with a purpose (GWAPs) (Von Ahn et al., 2006). Despite the good quality of collected assertions, manual efforts proved to be tedious and limited in relation to the size and diversity of the collected knowledge.

In light of the limitations of manual efforts, researchers shifted to large-scale commonsense knowledge acquisition by automatically harvesting textual resources. Moreover, the concurrent advancements in machine learning (ML) and information retrieval (IR) techniques, coupled with the abundance of textual resources on the Web, made the orientation towards automation even more appealing. Automatic methods leveraged on textual resources via pattern matching to discover potential valid assertions, followed by validation and/or scoring to filter the most plausible assertions. Some papers relied on hand-crafted extraction patterns (Pasca, 2014; Clark and Harrison, 2009; Etzioni et al., 2004), while others followed bootstrapping methods of patterns generation and facts extraction (Tandon and De Melo, 2010; Tandon et al., 2011). These methods have either populated a predefined knowledge base scheme or followed scheme-free open information extraction techniques. A limitation of automatic methods comes from the implicit and hard to articulate nature of commonsense knowledge. Therefore, despite the high recall and the expanded coverage of these method, they suffer from low precision.

To handle this, commonsense reasoning perform inference on existing knowledge to generalize beyond what is known. This direction of commonsense knowledge acquisition goes beyond literal extraction of explicit knowledge to elicitation of implicit assertions. Early Commonsense reasoning methods are basically logical models that fit mathematical models to existing knowledge. Logical rea-

soning is an insightful and powerful tool, however, the mathematical complexity might not scale well to the size of current knowledge bases (Chklovski, 2003).

By representing a knowledge base as a graph, a family of techniques referred to as knowledge graph embedding (KGE) converts knowledge graph entities and relations into k -dimensional vector representations that capture the inherent structure of the knowledge graph. To further enhance these representations, a series of models extended basic KGE models by incorporating different external information such as context, description, and entity types, in order to capture the semantic relatedness and semantic regularities associated with entities and relations.

The resulting representations are then utilized to perform reasoning over the knowledge graph. These methods deliver eminent performance in enriching encyclopaedic knowledge bases, such as DBpedia (Lehmann et al., 2015) and Freebase (Bollacker et al., 2008), with missing facts. Nevertheless, such performance is not observed when KGE models are applied to commonsense knowledge bases, mainly due to:

1. Commonsense knowledge is rather ambiguous and difficult to be matched in text, therefore, inducing semantic information directly from raw text could be a hurdle for text-enhanced KGE models, and subsequently limiting the effectiveness of semantic representations.
2. Commonsense concepts are abstract terms, therefore, it is not uncommon for a concept to have multiple meanings or senses. However, in most CSKBs, concepts are not disambiguated. Subsequently, knowledge graph embedding models and semantic distribution models will conflate the inherent structure and the lexical semantics of all the senses associated with a concept into a single vector representation. In this case, the resulting vector representation might not be able to capture all senses associated with the concept. Or it might get disrupted by all the senses such that it can not capture any.

In this thesis, we propose enhancements on knowledge graph embedding models aiming to improve their semantic representations. Our ultimate goal is to expand existing commonsense knowledge bases through augmenting them

with missing facts. Thus, the enhanced knowledge graph embedding models are tailored to improve commonsense reasoning. In particular, we propose two enhanced knowledge graph embedding models:

1. Semantically enhanced knowledge graph embedding models for commonsense knowledge acquisition.
2. Sense-disambiguated knowledge graph embedding models for commonsense knowledge acquisition.

1.2 Contributions

1. Semantically Enhanced KGE Models for CSKA

In this part, we advise an improved knowledge graph embedding model with the aim of enriching commonsense knowledge bases with new assertions. We propose a compositional approach that combines knowledge graph structural information with refined semantic information into a unified knowledge graph representation learning framework. The semantic information are meant to provide insight into concepts and relation meanings to compensate for the lack of explicit textual mention of concepts and semantic relations. This draws on the idea that importing semantically refined contextual information to commonsense knowledge graph representation learning will result in more focused embeddings without losing generalization capability. We use three different types of semantically refined context to incorporate into the model.

2. Sense Disambiguated KGE Models for CSKA

In this part, we propose an unsupervised model to learn various concepts' senses through analysing their contextual information in text corpus. We further expand commonsense knowledge bases by breaking down concepts to their corresponding senses, then learn sense-specific structural, contextual, and semantic embeddings for disambiguated concepts. These embeddings are then used for commonsense reasoning.

1.3 Challenges

Commonsense knowledge acquisition is a difficult task with unique challenges that stems from the characteristics of the knowledge itself. In this section, we review some of these challenges.

1. **Implicitness:** People view commonsense knowledge as default assumptions about everyday life that are assumed to be possessed by everyone; therefore, they often take it for granted and dismiss it from communication. Therefore, manual contributors find it difficult to think about and articulate what they take for granted and typical information extraction methods that depends on harvesting surface textual resources would face some difficulties in dealing with the implicitness of CSK. This urges for more advanced methods that can perform reasoning and inferencing to complement pattern-based extraction methods.
2. **Multimodality:** Unlike encyclopaedic knowledge which is mainly found in textual content, commonsense knowledge can be found in textual as well as visual content hence, multimodal approaches or composition models for knowledge acquisition are fundamental for expanding existing commonsense knowledge bases.
3. **Diversity:** Commonsense knowledge covers each and every aspect of our daily life and encompasses vast range of human knowledge. It can be generally characterised as being type and domain-independent. The involved concepts, phrases or relations can't be fully enumerated. The challenge facing acquisition process is the ability to tap on as much as possible of these diverse domains to retain generic CSK capable to serve general AI applications. Examples of such attempts is the shift from domain-specific corpora to general domain ones, or resorting to open information extraction approaches to go beyond restricted ontologies to extract all possible relations.
4. **Automation:** The generality and the universal scope of commonsense knowledge makes its acquisition a huge task that is beyond humans capacity to codify. Subsequently, it was necessary to shift from manual

approaches to automated and semi-automated ones. Specifically, reasoning approach aims to automatically infer new knowledge based on what is known through analogies and similarity. Mining approach can be fully automated when dealing with schema-free knowledge collection as in open information extraction, or semi-automated as in pattern-based bootstrapping methods.

5. Efficiency: With the advancements in computational performance, one would expect that the rate of CSK acquisition would increase equally, however, this is not the case. For mining approaches, the acquisition rate is often associated to the type and quality of the provided corpora as well as to whether the target is a fixed ontology or not. For reasoning approaches, as the size of existing knowledge increases, the efficiency of producing potential missing commonsense assertions would improve.
6. Need huge initial investments: In an interview (Dreifus, 1998), Marvin Minsky remarked that “Common sense is knowing maybe 30 or 50 million things about the world and having them represented so that when something happens, you can make analogies with others”.

1.4 Scope of Research

The focus of the thesis is to expand commonsense knowledge bases by predicting missing links among existing concepts. We adopt a vector space model reasoning approach to accomplish this goal. We pose the task of commonsense knowledge acquisition problem as knowledge base completion (KBC) task that is typically dependent on knowledge graph embeddings. We introduce two enhancements on GKE models by (1) incorporating auxiliary semantic informations to the KGE framework, and (2) learning multiple sense-specific embeddings per concept. Our study used a set of knowledge bases and information resources. We expand the English portion of ConceptNet commonsense knowledge base. We conducted two projects, each used a selected subset of ConceptNet. The filtering process for each subset will be described in details in respective sections. For auxiliary information, we use Numberbatch, AffectiveSpace, ProBase, Isacore, and word2vec word embeddings.

1.5 Thesis Outline

This report is organized as follows: Chapter 2 situates this research in the context of prior work. It first defines commonsense knowledge and review some of the commonsense knowledge bases. It then review various commonsense acquisition techniques. Chapter 3 present the proposed models while in chapter 4, we describes our datasets and experimental setups. In Chapter 5 we evaluate our method and discuss results. In Chapter 6 we conclude, summarizing what we have learned and offering suggestions for future work.

Chapter 2

Related Work

2.1 Background

2.1.1 Commonsense knowledge

Although there is no formal definition of commonsense knowledge, it can be roughly defined as a large collection of agreed-upon facts that are learned as a person grows up through daily life experiences. It spans unlimited range of domains including uses of objects, their properties, location and duration of events, urges and emotions of people, etc. It refers to the implicit knowledge that is shared among people and well known such that it is often dropped from communication, but is essential to carry out daily tasks. Some examples: phones are used to make calls, people use their teeth to chew food, people close their eyes when they sleep, etc. As per Zang et al. (Zang et al., 2013), commonsense knowledge can be defined by its characteristics as being shared by almost all people, fundamental and well understood that it is taken for granted, implicit, large-scale in both amount and diversity, open-domain that encompass all aspects of daily life, and default assumptions in typical situation that are open to exceptions.

In contrast to factual knowledge, commonsense is an ontological knowledge that is concerned with relations and properties of abstract concepts and classes rather than concrete entities or instances of these classes. Commonsense knowledge encompasses concepts and relation hierarchy which are enablers for commonsense reasoning and inference.

2.1.2 Commonsense Knowledge Bases

A knowledge base can be defined as a collection of assertions/facts that are gathered and represented as triples of the form $(head_term, predicate, tail_term)$, implying the existence of a labelled connection between two terms. In commonsense knowledge bases (CSKBs), terms correspond to abstract concepts (ontologies) rather than concrete instances of these concepts. A number of commonsense Knowledge bases has been constructed in the last three decades. The most prominent ones include Cyc (Lenat, 1995), WordNet (Miller, 1995) and ConceptNet (Liu and Singh, 2004). Most recently, Nicket Tandon has build WebChild (Tandon et al., 2017), a new fully automated commonsense knowledge base. We summarize the statistics of some CSKBs in table 2.1, then we describe them in more details:

Reference	Year	Source	Concepts	Relations	Assertions
Cyc (Lenat, 1995)	1984	Curated	500,000	17,000	7,000,000
ThoughtTreasure (Mueller, 1998)	1994	Curated	27,000	N.A	51,000
WordNet (Miller, 1995)	1995	Curated	155,327	~ 10	207,016
ConceptNet5.5 (Lenat, 1995)	2016	Semi-automated	1,803,873	38	28,000,000
WebChild 2.0 (Tandon et al., 2017)	2017	Automatic	2,300,000	6360	18,000,000

Table 2.1: Commonsense Knowledge Bases Statistics

2.1.2.1 Cyc

Cyc is the very first project to construct comprehensive commonsense knowledge bases started in mid 1980s and continued for 15 years. At the beginning, it was manually codified by group of skilled system experts in a formal predicate calculus like syntax language called (CycL). The commonsense knowledge in Cyc consist of facts, rules of thumb, and heuristics for reasoning about the objects and events of everyday life. By design, Cyc assertions have the property that they are true only

in certain contexts. Thus, Cyc’s assertions are organized in 20,000 micro-theories of shared assumptions. Cyc contains 500,000 terms, 17,000 relations, and around 7,000,000 assertions. In addition to the knowledge base, Cyc has a collection of inference engines to perform reasoning on its knowledge.

2.1.2.2 ThoughtTreasure

ThoughtTreasure (Mueller, 1998) is a commonsense knowledge base with an architecture for natural language understanding. Concepts in ThoughtTreasure has an upper ontology and several domain-specific lower ontologies. Further, each concept is associated with zero or more lexical entries (words and phrases). ThoughtTreasure contains 27,000 concepts linked to one another through 51,000 assertions. It also contains 35,000 English words/phrases, and 21,5000 French words/phrases.

2.1.2.3 HowNet

HowNet (Zhendong and Qiang, 2006) is an online linguistic commonsense knowledge base uncovering relationships between concepts or attributes of concepts. HowNet has more than 192,000 records which are represented with Knowledge Database Markup Language (KDML). Its concepts are denoted by words and expressions in both Chinese and English. These concepts are defined on the top of sememes, the smallest units of meaning. All sememes have been classified into four subclasses, including entity, event, attribute, and attribute-value; they are also organized into taxonomies respectively.

2.1.2.4 WordNet

WordNet is a handcrafted lexical database of English words which includes the lexical categories nouns, adjectives, verbs, and adverbs, and that is optimized for lexical categorization and word-similarity determination (Cambria et al., 2014). WordNet distinguish different senses of a word, where each sense is a distinct meaning that a word can assume, and group words with the same sense into sets of cognitive synonyms called ‘synsets’. In addition, each synset is associated with number indicating the frequency of its usage in text. Moreover, WordNet provides short definitions and usage examples of words, and count the frequency of relations between synsets or individual words. The latest version WordNet 3.1 contains 155,327 words organized in 175,979 synsets for a total of 207,016 word-sense pairs. The semantic relations in WordNet are between synsets rather than words and they are either linguistic

or commonsense relations. Example relations are synonym, hypernyms, hyponyms, substance meronym, etc. Nouns and adjective synsets are sparsely connected by **Attribute** relation (Tandon et al., 2014).

2.1.2.5 Open Mind Common Sense

The Open Mind Common Sense (OMCS) (Singh et al., 2002) is a project started in 1999 by the Common Sense Computing Initiative whose goal is to manually collect commonsense knowledge on a large scale. It relied on collaborative efforts of volunteers from general public to collect commonsense knowledge in the form of natural language statements which are then analysed to generate assertions. Since its launch in 1999, OMCS has accumulated over a million pieces of common sense information in English from over 15,000 contributors, in addition to extension to several other languages.

2.1.2.6 ConceptNet

ConceptNet (Lenat, 1995) is a huge semi-automated and multilingual commonsense knowledge resource, derived primarily from OMCS and other external resources, and represented in a WordNet inspired semantic network form. Its nodes are concepts expressed in natural language and its relations are extension of WordNet’s semantic relations ontology. A partial snapshot of actual knowledge in ConceptNet is given in Figure 2.1. ConceptNet has been revised and released with different versions starting from ConceptNet 2 and ending with the recent ConceptNet 5.5.

ConceptNet 5.5 (Speer et al., 2017) is the latest version of ConceptNet built from seven structured and unstructured knowledge resources (for more information, consult the original paper (Speer et al., 2017)). It contains over 21 million edges and over 8 million nodes from multilingual vocabulary and which are connected via 38 relations. Its English part consist of 1,803,873 concepts and around 28 million assertions. However, assertions are not well distributed among relation types with generic relations such as **RelatedTo**, **Synonym**, **IsA**, and **HasContext** constitute around 83% of instances while more specific relations such as **Causes**, **Desire**, **HasLastSubevent**, and **MotivatedByGoal** constitute as little as 1% of instances. Moreover, there are 83 languages in which it contains at least 10,000 nodes. ConceptNet5 relations are directed and also divided into symmetric and non-symmetric relations.

Figure 2.1: Snapshot of ConceptNet semantic network (Source: (Lenat, 1995))

2.1.2.7 WebChild 2.0

WebChild (Tandon et al., 2017) is a new semi-supervised semantically organized knowledge base. It was constructed by a series of algorithms to distill fine-grained disambiguated commonsense knowledge from massive amounts of text over multiple modalities. In particular, the knowledge base focused on three fine-grained commonsense knowledge categories: properties of objects, relationships between objects (comparative, part-whole), and objects interactions. The first version of WebChild (Tandon et al., 2014) associated sense-disambiguated nouns and adjectives over a set of 19 fine-grained relations indicating properties of objects such as **hasTaste**, **hasShape**, **evokesEmotion**, etc., where nouns and adjectives are disambiguated by mapping them onto their proper WordNet senses.

Their method started with collecting candidate assertions through automatically deriving seeds from WordNet and by pattern matching from web text collections. In particular, WebChild applied pattern matching over Google N-gram to collect assertions of *(noun, relation, adjective)* form, which are then filtered and disambiguated to become *(noun_sense, relation, adjective_sense)*. Each relation has a domain set of noun senses that appear as left-hand arguments, and a range set of adjective senses that appear as right-hand arguments. Label Propagation algorithm is then used to

serve two goals; one is providing domain sets and range sets for each relation, and second is providing confidence-ranked assertions between WordNet sense. Tandon N. followed this work with several adjustments to extract part-whole relations (Tandon et al., 2016), and activities (Tandon et al., 2015).

2.1.3 Knowledge Graph Embedding

2.1.3.1 Knowledge Graph

In recent years, the term “knowledge graph” has been frequently used to refer to graph-based knowledge representation and very often used interchangeably with the term “knowledge base”. It became popular after being reinvented by Google’s Knowledge Graph. Since then, it has been used loosely without a consensus on its formal definition. Ehrlinger and Wöß (Ehrlinger and Wöß, 2016) made an effort to collect some state-of-the-art definitions used in the literature and then proposed their own definition. A notable definition by Paulheim (Paulheim, 2017) opt to define knowledge graphs through some of their characteristics that distinguish them from merely graph-formatted data collection:

A knowledge graph (i) mainly describes real world entities and their inter-relations, organized in a graph, (ii) defines possible classes and relations of entities in a schema, (iii) allows for potentially interrelating arbitrary entities with each other and (iv) covers various topical domains.

More superficially, a knowledge graph is a multi-relational graph whose nodes correspond to entities and typed-edges correspond to relations between entities. Each edge represent a fact of form $(head_entity, predicate, tail_entity)$.

2.1.3.2 Knowledge Graph Embeddings

Knowledge Graph Embedding is defined as the task of learning contentious vector space representations for entities and relations of a knowledge bases such that the probability of having a *relation* connecting *head* and *tail* entities (denoted (h, r, t)) can be assessed through a score function characterized by the relation connecting the two entities $f_r(h, r, t)$. In other words, the main idea of these models is that relations between entities can be modelled as the interactions between their vector representations, where there are many ways in which these interactions can take place. These representations can be used in many tasks such as knowledge graph completion, link prediction, relation extraction, and so on. Different relation modeling methods

have been proposed in the literature and they mainly differ in the definition of the score function which is characterized by the way relation transformation operates. Additionally, a main focus of most of these methods is to reach the best trade-off between model’s expressivity and complexity to ensure it’s tractability over large scale knowledge graphs.

Formally, given a set of entities $\mathcal{E}$ and relations $\mathcal{R}$. A knowledge base $\mathcal{G}$ consist of set of triples (h, r, t) such that $h, t \in \mathcal{E}$ and $r \in \mathcal{R}$. Also, let’s denote the set of true triples (h, r, t) that belong to $\mathcal{G}$ by Δ and incorrect triples $\{(h', r, t) | h' \in \mathcal{E}, (h', r, t) \notin \mathcal{G}\} \cup \{(h, r, t') | t' \in \mathcal{E}, (h, r, t') \notin \mathcal{G}\}$ by Δ' . The embedding models learn entities and relations representations by optimizing a global loss function over all facts such that these representations encode local connectivity patterns, hence helping to reason new facts by generalizing over existing ones. A margin-based loss function is commonly used in these models.

The earliest approach targeting multi-relational data is the energy based model Structured Embedding (SE) proposed by Bordes et al. (Bordes et al., 2011). The model learns $\mathbb{R}^k$ vector representations per entity and two $\mathbb{R}^{k \times k}$ projection matrices; i.e. $\mathbf{W}_{\mathbf{r}, \mathbf{h}} \in \mathbb{R}^{k \times k}$ and $\mathbf{W}_{\mathbf{r}, \mathbf{t}} \in \mathbb{R}^{k \times k}$, per relation. The model then projects the head and tail entities of a triple into a common subspace through the two relation-specific matrices and scores a triple (h, r, t) by the distance between the entities’ projections $f_r(h, r, t) = \|\mathbf{W}_{\mathbf{r}, \mathbf{h}}\mathbf{h} - \mathbf{W}_{\mathbf{r}, \mathbf{t}}\mathbf{t}\|$ such that the distance is small for correct triples and large for corrupted ones. The two matrices per relation are meant to account for possible asymmetry in relationships. One weakness of this model stems from that the use of two separate matrices does not allow direct interactions between entities, and instead model the interactions between their projections, hence making SE unable to precisely capture the interaction between entities.

Bordes et al. proposed another embedding model TransE (Bordes et al., 2013) inspired from the successful word2vec language model by Mikolov (Mikolov et al., 2013b). TransE represents a relationship r as a translation between the vector representation of two entities h and t ; that is, if triple (h, r, t) holds, then the embedding of the entity t is close to the translation of the entity h by the relation r (i.e $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$), where the score function is defined as the distance $f_r(h, r, t) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_{1/2}$, and the distance function is the first or second norm. Despite the model simplicity and reduced number of parameters (efficiency), its predictive performance showed noticeably improved results over previous methods especially when dealing with one-to-one relations, however, it does not do well in dealing with relations of different mapping properties such as one-to-many, many-to-one, many-to-many, etc.

To overcome TransE flaw, a new model TransH (Wang et al., 2014a) enabled entities to have different representations when involved in different types of relations by moving the translation operation from entity embedding space to relation-specific embedding space. The model thus regards a relation as hyperplane characterized by its norm $\mathbf{w}_r$ and a translation vector $\mathbf{d}_r$ on that hyperplane. Under this model, a triple (t, r, h) is a translation operation $\mathbf{d}_r$ of the two entities' projections $\mathbf{h}_\perp$ and $\mathbf{t}_\perp$ on the relation hyperplane $\mathbf{w}_r$. The score of a triple then becomes $f_r(h, r, t) = \|\mathbf{h}_\perp + \mathbf{d}_r - \mathbf{t}_\perp\|_2^2$. This interpretation of relation improved results in reflexive, one-to-many, many-to-one, and many-to-many relations without a significant increase in model complexity.

As pointed out by Yankai et al. (Lin et al., 2015b), however, some weaknesses in the expressivity of TransE and TransH is that both models embed entities and relations in the same $\mathbb{R}^k$ space, while they are of different types and should thus be embedded into different spaces. For example, entities may have multiple aspects in which they can be similar in some of these aspects in particular relations and dissimilar in other relations. TransR (Lin et al., 2015b) propose embedding entities and relations into distinct entity and relation spaces $\mathbb{R}^k$ and $\mathbb{R}^d$ respectively. It then define projection matrix $\mathbf{M}_r \in \mathbb{R}^{k \times d}$ to obtain relation-specific entities projections $\mathbf{h}_r = \mathbf{h}\mathbf{M}_r$ and $\mathbf{t}_r = \mathbf{t}\mathbf{M}_r$. Triples are defined as translation between the projected entities with as corresponding score function $f_r(h, r, t) = \|\mathbf{h}_r + \mathbf{r} - \mathbf{t}_r\|_2^2$. TransR has significant improvements compared with previous state-of-the-art models.

A non-linear class of relation transformation was introduced in Single Layer Model (Socher et al., 2013b) which borrowed ideas from the text embedding models in which $\mathbf{h}$ and $\mathbf{t}$ are concatenated and fed as input to a neural network with non-linear hidden layer and linear output layer were triples is scores as $\mathbf{u}^T f(\mathbf{W}_{r,h}\mathbf{h} + \mathbf{W}_{r,t}\mathbf{t} + \mathbf{b}_r)$. The TNT model further extend the this work by adding the second-order entities correlation into the input layer such that the score function become $\mathbf{u}^T f(\mathbf{h}^T \mathbf{W}_r \mathbf{t} + \mathbf{W}_{r,h}\mathbf{h} + \mathbf{W}_{r,t}\mathbf{t} + \mathbf{b}_r)$.

2.1.3.3 Joint Text and graph embedding models

Text embedding (2.1.4) and knowledge embedding models have their strengths and limitations individually that would be complementary to each other when combined. For example, knowledge embedding learns representation of entities/relations that exist in a KB, and thus its capability is limited to predicting missing facts between existing entities. Text models in the other hand are able to extract new facts from text, for most of which, the relation connecting words/phrases is unknown. Recent work attempted to combine these two models in joint framework to better improve

the results of knowledge base completion. This class of models utilize information that can be induced from structured data in knowledge bases and information that can be induced from unstructured data sources such as text corpora or entities' and relations' descriptions.

Methods under this umbrella follow one of two main paradigms. One is learning words and entities embedding jointly into a unified vector space. Training these models is a burden due to the computation complexity required to deal with the size of entities and vocabularies (Toutanova et al., 2015; Han et al., 2016; Wu et al., 2016). The other paradigm learn words embedding and entities embedding separately followed by applying annotation or linking algorithms to align text to entities, after which the two embeddings are joined in a particular manner (Wang et al., 2014a; Yamada et al., 2016).

2.1.4 Semantic Distributional Models

Word embedding models refers to the collection of algorithms and techniques in natural language processing that maps words and phrases in vocabulary to compact low-dimensional vector representations such that these representations capture semantic and syntactic information of individual words. Word vectors can be useful for a variety of applications such as information retrieval (Manning et al., 2008), document classification (Sebastiani, 2002), question answering (Tellex et al., 2003), named entity recognition (Turian et al., 2010), and parsing (Socher et al., 2013a). Different models performed this word to vector mapping including (1) Latent Semantic Analysis (LSA), (2) Latent Dirichlet Allocation (LDA), and (3) Neural Networks (NN). The first two models fall under *global* matrix-factorization scheme that accounts for global co-occurrence statistics. They perform low-rank approximations to decompose large matrices that capture statistical information about a corpus. Neural Network models, on the other hand, utilize *local* context-window methods. In general, these models are trained to optimize generic objective functions measuring syntactic and semantic word similarities.

The earliest attempts to use neural network for learning words vector representations are dated back to mid 1980s were done by Rumelhart et al. (Rumelhart et al., 1986) and Hinton et al. (Hinton et al., 1986). Lately, Mikolov et al. (Mikolov et al., 2013b; Mikolov et al., 2013a) introduced two highly efficient log-linear models, continuous bag of words (CBOW) and continuous skip-gram (SG), to produce a dis-

tributed representation of words from huge datasets. The continuous bag-of-words (CBOW) model predicts the current word from a window of surrounding context words. The order of context words does not influence prediction (bag-of-words assumption). Specifically, context words are projected to their embeddings and then summed. Based on the summed embedding, log-linear classifiers are employed to predict the current word. Formally given a sequence of training words $w_1, w_2, \dots, w_T$, and given a window size c such that there are c words to each side of a target word, the CBOW model learn word embedding by maximizing the objective function:

$$\frac{1}{T} \sum_{t=1}^T \log p(w_t \mid \sum_{-c \leq j \leq c, j \neq 0} w_{t+j}) \quad (2.1)$$

The skip-gram model on the other hand uses the current word to predict the surrounding window of context words. The skip-gram architecture weighs nearby context words more heavily than more distant context words. Here, the current word is projected to its embedding, and log-linear classifiers are further adopted to predict its context. Formally, the skip-gram model learn word embedding by maximizing the objective function:

$$\frac{1}{T} \sum_{t=1}^T \sum_{-c \leq j \leq c, j \neq 0} \log p(w_{t+j} \mid w_t) \quad (2.2)$$

Denoting a target word as w_t and its embedding as v_{w_t} , and denoting context as w_c and its embedding as v_{w_c} , skip-gram define the probability $p(w_c \mid w_t)$ as a Softmax function:

$$p(w_c \mid w_t) = \frac{\exp(v_{w_c}^T v_{w_t})}{\sum_{w=1}^W \exp(v_w^T v_{w_t})} \quad (2.3)$$

For CBOW, w_t and w_c as well as their embeddings are swapped. However, softmax is impractical because the cost of computing the gradient is proportional to the vocabulary size W . An alternative and efficient formulation that was proposed in (Mikolov et al., 2013b) is negative sampling which posits that a good model should be able to differentiate data from noise by means of logistic regression. Formally, negative sampling is defined by the objective

$$\log \sigma(v_{w_c}^T v_{w_t}) + \sum_{i=1}^k \mathbb{E} \sim P_n(w) [\log \sigma(-v_{w_i}^T v_{w_t})] \quad (2.4)$$

k is a hyper-parameter that specifies the number of random negative samples to use in contrast to the positive pull between the target and the context and that are sampled from a noise distribution $P_n(w)$.

In addition to models’ efficiency, word2vec introduced a new evaluation scheme that is based on words analogies and syntactic and semantic regularities. For example, the skip-gram model can learn word embedding such that vectors of word pairs that share same relations are almost parallel without knowing the exact relation between the word pairs, instead the relation is characterized by a relation-specific word vector offset (Mikolov et al., 2013c; Zhila et al., 2013), e.g., $vec(\text{Italy}) - vec(\text{Rome}) \approx vec(\text{France}) - vec(\text{Paris})$.

The global and local model families for learning word embeddings have their own strengths and shortcomings. While the first is able to exploit the statistical information encoded in global word co-occurrences, the second is able to capture fine-grained similarities and regularities in words semantics. Pennington et al. (Pennington et al., 2014) constructed GloVe model that combines the benefits of both models by exploiting the global statistical information of matrix factorization methods while simultaneously capturing the meaningful linear substructures prevalent in recent log-bilinear prediction-based methods like word2vec.

A body of work extended word embedding to context embedding with the aim to capture the inter-dependence between a target word and its surrounding context. One approach is Average-of-Word-Embeddings AWE, in which context words’ stand alone embeddings are averaged or weight-averaged. The drawback of AWE is that correlation between words is not captured. Context2Vec (Melamud et al., 2016) is another model that learns a generic task-independent embedding function for variable-length sentential contexts around target words simultaneously while learning target word embedding, with the objective of having the context predict the target word via a log-linear model. It uses two bidirectional LSTM recurrent neural network to learn two separate left-to-right and right-to-left order preserving context embeddings, then concatenate the two embeddings. The context and target word embeddings are passed to MLP to learn non-linear dependencies.

2.2 Building Commonsense Knowledge Bases

Building a representative commonsense knowledge base that can be useful for AI tasks is not a straightforward process. It requires the involvement of multiple techniques,

methods, and resources. In this section, we categorize approaches into three main types based on the main technique of knowledge acquisition: manual approaches, text mining approaches, and reasoning approaches. In many cases, however, CSKBs are acquired by multiple techniques and from multiple resources.

2.2.1 Manual Acquisition

The earliest stages of commonsense acquisition relied on manual efforts to collect and codify commonsense assertions. These efforts can be divided mainly into two types, Labor commonsense acquisition and Collaborative commonsense acquisition. We review these two types in more details.

2.2.1.1 Labor Commonsense Acquisition

At beginning, researchers relied on teams of either paid system experts and knowledge engineers to codify commonsense entries in a formal language that is readable by computers, or unpaid and untrained volunteers to write commonsense entries as natural language sentences which will be examined and converted to formal language by knowledge engineers, or to verify knowledge entered by other contributors.

The first stage of Cyc (Lenat, 1995) construction consisted of manually codifying millions of assertions and inference rules in CycL language totally by ontologists and knowledge Engineers. These assertions are of types that are believed to unlikely be expressed in textual resources. In another setting, Cyc utilized volunteers rather than specialized experts to enter straightforward and easy to formalize commonsense knowledge such as "Fishes can swim" (Witbrock et al., 2005). Practically, volunteers are allowed to enter these facts through user-friendly interfaces in which they are able to either fill blanks in natural language or select among plausible choices. Facts in natural language are then converted to formal language, after which, they are filtered and verified according to their compatibility (or compliance) with existing knowledge or presence of grounding evidence in external corpus, in addition to voting by trusted reviewers.

ThoughtTreasure was also manually created by Erik Mueller (Mueller, 1998) beginning from 1994 as a platform for natural language processing and commonsense reasoning. ThoughtTreasure contains both a knowledge base and natural language understanding tools. The knowledge base stores both declarative and procedural concepts where concepts are connected to each other by statements.

WordNet (Miller, 1995) and HowNet (Zhendong and Qiang, 2006) are another

two manually created resources that are basically meant as linguistic commonsense knowledge bases. WordNet development was started in 1993 by a group of researcher at Princeton University, and HowNet started in 2006 as a Chinese-English bilingual commonsense knowledge base.

2.2.1.2 Collaborative Commonsense Acquisition

To scale up the labor-intensive manual process, researchers turned to collaborative efforts through public platforms, such as crowdsourcing or games with a purpose (GWAPs). These platforms adopted interactive approach with users to keep them engaged. For example, users may receive real-time feedback of the quality of their entries, giving them the sense that computer is understanding them, thus feeling the enthusiasm to continue with knowledge entry. In the following, we describe some of these collaborative efforts.

Interactive tools: Cyc project utilized lightly trained Subject Matter Experts (SME) to expand specific domain knowledge through KRAKEN (Panton et al., 2002), an interactive tool that facilitate natural language interactions with the SME. KRAKEN was designed as a natural-language based conversational interface between SMEs and Cyc KB, which translates back and forth between English and the KBs logical representation language.

Open Mind Commons (Speer, 2007) is an interactive interface for collecting commonsense knowledge from volunteers, which supply users with feedback on the knowledge they enter. Feedback helps not only retain users interest, but also results in higher-quality and more relevant entries. The system perform analogical inference based on the knowledge that it already has on a topic, to come up with a set of potential commonsense statements. These statements are then presented to users to either confirm or reject. For example, the system may prompt a user with a question like “*A bicycle would be found on the street. Is this common sense?*” to which the user can answer with *Yes* or *No* to confirm or reject. If a user answered a question with a *No*, the system will ask the user to change an item to make the statement true. This process serves multiple goals; it confirms to the user that the system is understanding and learning from the data it acquires, helps to fill in gaps in a given topic area and make knowledge base more strongly connected, and evaluates inference methods correctness. Another interface present users with fill-in-the-blank questions that are derived in similar procedure: simply finds inference candidates with one object left unknown. For example, the system may ask “*You are likely to find _____ in a su-*

permarket.”. This, too, helps to make the knowledge in the database more strongly connected. The feedback that users receive include new inferences and analogies that have been made on the basis of their new contributions, ratings of their contributions by other users, and follow-up questions that the system asks after a user rejects a potential inference.

Crowdsourcing: Crowdsourcing, as first defined by Jeff Howe and Mark Robinson (Howe, 2006), “represents the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call”. AI researchers picked up on this concept in the context of commonsense acquisition. In the project Open Mind Common Sentic (Cambria et al., 2012b), Cambria et al. transformed the process of manually entering affective commonsense knowledge into an enjoyable activity through a crowdsourcing platform, that follow the methods of Open Mind Commons (Speer, 2007) in which volunteers over the Web are challenged through mood-spotting and fill-in-the blank questions. In mood-spotting, users are urged to select an emoticon according the overall affect they can infer from a given sentence, while in fill-in-the blank questions, users are to complete sentences such as “*opening a Christmas gift makes feel _____*”.

Games with a purpose: Games with a purpose (von Ahn, 2006) are a collective intelligence approach based on the general research paradigm, Human Computation, which envision harnessing human brainpower made available by multitudes of casual gamers to perform tasks that, despite being trivial for humans to compute, are rather challenging for even the most sophisticated computer programs. Developers of AI applications tapped on this idea to collect commonsense knowledge. GWAPs have advantage over the volunteer-based efforts in that rather than relying on willingness of unpaid volunteers to contribute their time and knowledge, GWAPs provide an enjoyable gameplay experience that is typically designed with incentives (win a game/score more) to keep players engaged while having fun, in addition to mechanisms to verify the correctness of collected knowledge.

Cyc project developers built FACTory Game (Lenat and Guha, 1989) in which they ask players to judge commonsense statements that are generated from the CYC repository as being true, false, or non-sense in addition to a don’t-know option to abstain. FACTory Game reward players with points upon agreeing with the majority answer for a fact and a certain consensus threshold has been reached. With a similar

principle, Concept Game (Herdagdelen and Baroni, 2010), verify candidate common-sense facts collected through pattern-based text mining. Concept Game was build with the purpose to expand a commonsense repository, rather than just verifying its existing knowledge, by filtering and verifying text-mined candidate assertions. Such approach alleviates the difficulty of recalling and defining commonsense knowledge by human contributors and filter the noisy text-mining based extractions. Concept Game present players with candidate assertions in a slot-machine fashion and allow players to validate those assertions while they play and award them for true positives while penalize them for false positives.

Verbosity (Von Ahn et al., 2006) is a word-guessing interactive game for collecting common-sense facts in order to train reasoning algorithms. Given a concept word, the game aims to collect commonsense facts about the concept through a set of hint sentences. The game work as follows: two randomly selected players keep alternating roles in which one is a narrator and the other is a guesser. The narrator is given the concept “secret” word and provide hints to the guesser using sentence templates that describe the word without using the word itself, while the guesser has to guess the word in the shortest time possible. The narrator also help the guesser by scoring answers as “hot” or “cold”. For example, given the word “squirrel”, and hint sentences like “*it is a type of tree rodent*” and “*it looks like chipmunk*” establish the commonsense facts “*squirrel is a type of tree rodent*” and “*squirrel looks like chipmunk*”.

Common Consensus (Lieberman et al., 2007) is an online self-sustaining game, designed to collect and validate a specific type of commonsense knowledge, namely, knowledge about everyday goals. The knowledge collected from this game help recognize goals from actions or conclude a sequence of actions leading to goals. It also associate goal with sub-goals, parent-goals, analogous goals, motivations, and situations. In the game, players are presented with open ended questions about a goal, and are encouraged to answer with what they expect an anonymous person would say. The players are then rewarded based on the commonality of their answers. For example, for the goal “*book a flight*”, the game can collect actions to achieve the goal from answers to the question “What are some things you would use to *book a flight*?”, or motivations leading to the goal from answers to the question “Why would you want to *book a flight*?”.

Kuo et al. (Kuo et al., 2009) presented two community-based games to collect commonsense knowledge in Chinese, deployed on two leading online social platforms. The games operate in two interaction modes; direct interaction mode or indirect

interaction mode. Rapport Game on Facebook harvest direct interactions between players to construct a semantic network that encodes common-sense knowledge. In this game, players either construct commonsense facts by filling subject or object place-holders of OMCS sentence templates such as “A likes B”, or validate filled assertions. Virtual Pet is a pet-raising game on PTT, a famous bulletin board system in Taiwan, that depends on indirect interactions between players through their pets to answer commonsense questions. Players take care of their pets’ in many ways some of which are feeding them or helping them become more intelligent through gaining commonsense points. Players can ask or answer their pets’ questions to gain commonsense points. When a player ask a question, that question would be answered by another player. These games collected over 500,000 verified statements which have become the OMCS Chinese database.

Hourglass Game was developed as a part of the Open Mind Common Sentic project (Cambria et al., 2012b) which perform affective commonsense knowledge acquisition. Affective commonsense associate concepts with related, contained, or produced affective emotions. Hourglass Game present players with affective concepts and ask them to choose, from Hourglass emotion categorization model 2.2, the sentic level associated with the presented concepts. The players are awarded based on the accuracy of their associations and their speed in creating affective matches. The game also collect new affective commonsense knowledge by aggregating information of random multi-word expressions that are not previously associated with any affective information.

GECKA (serious game engine for common-sense knowledge acquisition) (Cambria et al., 2015b) is a game engine for commonsense knowledge acquisition that aims to overcome the main drawbacks of traditional data-collecting games by empowering users to create their own GWAPs and by mining knowledge that is highly reusable and multi-purpose. To this end, GECKA offers functionalities typical of role-play games (RPGs), e.g., a question/answer dialogue box enabling communication and the exchange of objects (optionally tied to correct answers) between players and virtual world inhabitants, a library for enriching scenes with useful and yet visually-appealing objects, backgrounds, characters, and a branching storyline for defining how different game scenes are interconnected.

Figure 2.2: Hourglass of Emotions (Source:(Cambria et al., 2012a))

2.2.2 Mining-Based Acquisition

A shift toward large-scale commonsense knowledge acquisition leveraged on textual resources via pattern matching to discover potential valid assertions. Although curated resources have the advantage of having high precision, they tend to lack sufficient coverage. On the other hand, text mining techniques produce huge knowledge collections, however, at the cost of low precision, in addition to being limited to the knowledge that is expressed in explicit manner and which is amenable for data mining. Some papers relied on handcrafted extraction patterns (Pasca, 2014; Clark and Harrison, 2009; Etzioni et al., 2004), while others followed bootstrapping method of patterns generation and facts extraction (Tandon and De Melo, 2010; Tandon et al., 2011).

2.2.2.1 Semi-Automated

In semi-automated mining approaches, human contribution is present in either creating extraction patterns or validating and filtering resulting assertions.

As mentioned earlier in 2.1.1, ConceptNet is a semi-automatically created resource that was originally build as the semantic network representation of the knowl-

edge collected from OMCS projects, and that was later expanded from other external resources. In ConceptNet-2 (Liu and Singh, 2004) a three phase extraction process was applied to extract around 30,000 concepts and 1.6 million assertions from the 700,000 semi-structured English sentences of the Open Mind Common Sense Project. The extraction phases of this process consisted of applying approximately 50 hand-crafted extraction rules to the OMCS corpus to extract binary predicates. The extraction rules are regular expressions with syntactic and semantic constraints over predicates' arguments (concepts). Concepts involved in assertions are restricted with syntactic structure which is composed of combinations of four syntactic constructions: verbs (e.g. 'cook', 'run'), noun phrases (e.g. 'green dress', 'big house'), prepositional phrases (e.g. 'in office', 'at school'), and adjectival phrases (e.g. 'very hot', 'sweet'). The syntactic constraints also enforced restriction on the order of these components. Normalization phase followed by relaxation phases were then followed in order to reduce concepts to their canonical 'lemma' form and to smooth over semantic gaps and improve the connectivity of the network respectively.

A similar, yet, simpler pattern matching approach was applied to construct ConceptNet-3. Traditionally, regular expressions pattern-matching and chunking are used to translate the unparsed English sentences of Open Mind corpus into ConceptNet assertions. For example, an instance of **HasSubevent** relation can be recovered using a regular expression pattern like "One of the things you do when you (.+) is (.+)". Given the statement "One of the things you do when you drive is steer", for example, this would produce the predicate (*drive*, **HasSubevent**, *steer*). This method has its limitations however, such as producing incorrect extractions or certain patterns are impossible to recover. ConceptNet-3, thus, resorted to a simple parser as a kind of pattern matcher. Instead of matching with regular expressions, parser matches with place-holder phrases. The parser output two text strings and determines the plausibility of them being related. The produced raw predicates are then passed to a normalization process to determines which two concepts the two text strings correspond to, turning the raw predicate into a true edge of ConceptNet.

Eslick (Eslick, 2006) presented ConceptMiner semi-automated knowledge acquisition systems. The system employs extraction patterns and makes use of the knowledge in ConceptNet to extract commonsense knowledge from the web. It use some ConceptNet relation instances as seeds to derive general extraction patterns from the Web, then search the Web using these patterns to extract new relation instances in a

bootstrapping fashion. For example, a relation instance such as (dog, **DesireOf**, attention) derives search results such as *My/PRP dog/NN loves/VBZ attention/NN./* which in turn can be generalized into pattern of the form: $\langle X \rangle/NN \text{ loves}/VBZ \langle Y \rangle/NN$. This pattern is then used to extract potential relation instances from the Web. The extracted instances go through a sequence of filters to discriminate bad ones.

Pasca (Pasca, 2014) considered Google query log as a source of commonsense lexicalized assertions and used a set of manually specified patterns to recover commonsense knowledge. For example, they use patterns like *why [is | was|were]/[a|an|the|[nothing]]* to recover queries like *why are (cars) (made of steel)* or *why is a (newspaper) (written in columns)*. Queries returned by pattern matching are scored: $score(F, C) = LowBound(Wilson(N_+, N_-))$, where the fact is F , and C is a class (subject).

All sub-tasks of WebChild required extraction of candidate assertions. Tandon et al. presented an automatic approach for collecting entities from web content and deployed their method to build a large commonsense knowledge base called WebChild (Tandon et al., 2014). The knowledge base is focused on associating sense-disambiguated nouns and adjectives over a set of 19 fine-grained relations such as hasTaste, hasShape, evokesEmotion, etc., where nouns and adjectives are disambiguated by mapping them onto their proper WordNet senses. The method starts with collecting candidate assertions through automatically deriving seeds from WordNet and by pattern matching from web text collections. In particular, WebChild applied pattern matching over Google N-gram to collect assertions of $(noun, relation, adjective)$ form, which are then filtered and disambiguated to become $(noun_sense, relation, adjective_sense)$. Each *relation* has a domain set of noun senses that appear as left-hand arguments, and a range set of adjective senses that appear as right-hand arguments. Label Propagation algorithm is then used to serve two goals; one is providing domain sets and range sets for each relation, and second is providing confidence-ranked assertions between WordNet sense. Tandon N. followed this work with several adjustments to extract part-whole relations (Tandon et al., 2016), and activities (Tandon et al., 2015).

2.2.2.2 Automated

Traditional automatic information extraction (IE) systems recover all possible relational tuples concerning predefined set of target relations from labelled training set.

These methods take relations along with automatically induced or hand-crafted extraction patterns and match them over large-scale corpora. However, they do not scale to the web size, plus it is hard to define all relations in advance. Another IE paradigm, known as open information extraction (OIE) introduced by Banko et al. in 2007 (Banko et al., 2007), capture all possible assertions from open corpora without pre-specified extraction targets. These methods are relevant to commonsense knowledge in the sense that commonsense relations are diverse and can not be fully pre-specified. However, OIE results on redundant extractions that refer to the same assertions with different wordings. This would greatly hinder reasoning by lacking enough representation of each relation. Moreover, OIE does not distinguish between factual and commonsense knowledge.

TEXTRUNNER (Banko et al., 2007) is the first Web scale Open IE system. It perform a single scan of an open corpus to extract all possible tuples of form (*noun_phrase*, *relation_phrase*, *noun_phrase*) in a process that consist of three-stages: (1) a single-pass extractor: makes a single pass over the entire corpus to extract all candidate tuples. It starts by identifying all pairs of noun phrases (NPs) in the corpus using a chunker. These noun phrases are considered as entities, and the text between them is elicited to extract relation, phrases with heuristics to discard unlikely relations. (2) a self-supervised Naive Bayes classifier trained with unlexicalized part-of-speech (POS) and noun phrase features, to assess and retain tuples extracted in the previous step according to a trustworthiness measure (3) a redundancy-based assessor which assigns a probability to each retained tuple based on a probabilistic model of redundancy in text. When tested on a corpus of 9 million Web documents, TEXTRUNNER extracted 7.8 million well-formed tuples which are assertions like (*Edision*, *invented*, *light bulbs*), with accuracy 80.4%.

The heuristic approach of TEXTRUNNER results on some extractions that are rather incoherent or uninformative. REVERB (Fader et al., 2011) takes a step to eliminate the possibly of such undesired output by enforcing syntactic and lexical constraints on the verbal expression of binary-relation phrases. The Syntactic constraint eliminate meaningless relation extraction by matching relation phrases to POS tag patterns such that the captured relations are expressed in verb-noun combinations including light verb constructions. In particular, The syntactic constraint choose relation phrases that are either a simple verb phrase, a verb phrase followed immediately by a preposition or particle, a verb phrase followed by a simple noun

phrase and ending in a preposition or particle, or a concatenation of them in case multiple adjacent sequences are matched. Lexical constraints are then applied to retain relation phrases that have acceptable distinct argument support. To achieve this, REVERB parse POS-tagged and NP-chunked input sentences searching for the longest verb-started sequence of words satisfying the syntactic and lexical constraints and consider it as the relation phrase]. It then search for NP pairs surrounding extracted relations to form $(NP, relation_phrase, NP)$ tuples. The resulting extractions are then assigned a confidence score using a logistic regression classifier trained on set of features derived from the aforementioned constraints.

REVERB developers remarked that a large majority of extraction errors by Open IE systems come from incorrect or improperly-scoped arguments. For example, they assumed that arguments are simple noun phrases (NPs), disregarding more complicated arguments' structures such as NPs with prepositional attachments, lists of NPs, independent clauses, etc. Experiments on REVERB showed that 65% of errors had correct relation phrase but incorrect arguments, thus supporting the previous claim. Subsequently, they developed argument learning system termed ARGLEARNER to identify arguments given a sentence and relation phrase pair. ARGLEARNER uses multiple supervised statistical classifiers to first identify the relation phrase arguments that go beyond just noun phrases, and then to detect the left bound and the right bound of each argument. The classifiers use heuristic features include those that describe the noun phrase in question, context around it as well as the whole sentence, such as sentence length, POS-tags, capitalization and punctuation. The combination of REVERB relation phrases and ARGLEARNER arguments is named R2A2 (Etzioni et al., 2011).

Weltmodell (Akbik and Michael, 2014) is a commonsense knowledge base that was automatically generated from the dependency parse fragments of Google's syntactic N-Grams dataset. The dataset contains over 10 billion syntactic n-grams, which are rooted syntactic dependency tree fragments (noun phrases and verb phrases). Each tree fragment is annotated with the dependency information, its head word, and the frequency with which it occurred. Weltmodell applies the rule-set open-domain Information Extraction method described by (Akbik and Löser, 2012) on the dependency trees that contain verbs and all of its fragments, to collect subjects, particles, negations, passive subjects, direct and prepositional objects of the verb. Heuristic are then applied to standardize and arrange the arguments of collected facts in form

of statements with concept place-holders. The strength of the association between a statement and a concept is computed using PMI and marked the confidence in facts.

To more effectively harness textual resource to extract general knowledge, it is required to tap on the data lying at a level beneath the explicit content. This observation by Schubert led to the development of KNext system(Schubert, 2002), which derive implicit CSK in form of *general possibilistic propositions* from the textual corpus Penn Treebank. Here, *general* means that the relations are not predetermined specific kind of facts such as part-whole or causality, and *possibilistic* means the assertions are possible in the world, or, under certain conditions, implied to be normal or commonplace in the world. For example, given the sentence “he entered the house through its open door”, they can infer that “it is possible for a male to enter a house”, “houses probably have doors”, “doors can be open”, etc. KNext starts with matching general phrase structure to extract sub-trees from the Penn Treebank. For each successfully matched sub-tree, the system first abstracts the interpretations of each essential constituent of it, e.g., “an open window at the rare end of the car” would be abstracted to “a window”. After that, compositional interpretive rules help combine all abstracted interpretations and finally derive a general possibilistic proposition.

The OpenIE systems do not discriminate between encyclopedic and commonsense knowledge. This is partially because the arguments and relations are not canonicalized. These systems are typically not designed to construct and organize a commonsense KB (or even a KB), rather their goal is to acquire triples for a use-case like question answering.

2.2.3 Reasoning Based Acquisition

Commonsense reasoning is the process that allow humans to behave and interact based on their knowledge, experiences, beliefs, and even uncertainties (Anderson et al., 2013). It is the central part of human intelligence that allows them to perform and interact in all life situations. From AI perspective, commonsense reasoning aims to help computers build an understanding of human world and human reasoning behaviour such that it can behave and interact in a more human like manner. To enable the development of AI intelligence, we need to explicitize and transfer human knowledge as starting point . In the context of commonsense acquisition, reasoning models perform automatic knowledge acquisition by making rough guesses of valid

assertions based on existing knowledge.

Under the umbrella of KBC, vector space models learn entity and relation vector representations and use those representations to predict missing facts or to validate existing knowledge. There are few recent attempts to use vector representations of concepts and relations for the task of commonsense knowledge acquisition. The work in this direction is often focused on improving concept vector representations by incorporating external sources of information with salient features to capture the semantics of these concepts.

Aside from knowledge acquisition, Chen et al. (Chen et al., 2015) introduced some enhancements on concept representations learning that can be utilized in knowledge acquisition framework or any other semantic similarity and relatedness tasks. They suggested an extension of the well-known CBOW model to obtain better vector representation of concepts. The basic idea is that using semantically salient context rather than just general context will improve the quality of embeddings to reflect semantic proximity. Authors relied on word definitions and synonyms as well as lists and enumerations as contexts. Generated vectors were evaluated through word relatedness and story completion tasks. For word relatedness, they measured the similarity between words using Spearman's coefficient and compared the results with human judgment. A vivid conclusion of this paper is that different information sources and extraction methods can bring different sorts of information to concepts latent vectors. In this paper, the new information are definitions and list, subsequently the improvement in semantic similarity is naturally captured.

Chen et al. followed up by a statistical relation learning model for commonsense knowledge acquisition. In (Chen et al., 2016), authors presented a new approach for harvesting commonsense knowledge that relies on joint learning model from web-scale data. The model learn vector representations of commonsensical words and relations jointly using large-scale web information extractions and general corpus co-occurrences. The approach start by applying a pattern-based information extraction to acquire a large amount of commonsense knowledge in the form $(subject, predicate, object)$ triples. The model then learns words representations of subject and object by optimizing word2vec CBOW objective $\sum_{(w)} \log P(w|C(w))$ to capture general word co-occurrence information, where w denotes a word token in a large corpus and $C(w)$ denotes the word's context, and the model aim to learn word vectors $\mathbf{v}_w$ that maximizes the objective. The model simultaneously optimize for modeling the explicit relationships mined earlier. Denoting each mined relation

as (s, r, o) , where s, r , and o corresponds to subject, relation, and object respectively, the optimization function is $f_r(s, r, o) = \mathbf{v}_s^T \mathbf{M}_r \mathbf{v}_o$, where $\mathbf{v}_s$ and $\mathbf{v}_o$ are the word vectors for s and o and $\mathbf{M}_r$ is a matrix for relation r . Finally, vector representations are learned both from the relations and using the word2vec CBOW objective through a joint loss function of the two objectives.

Li et al. (Li et al., 2016) aimed to enrich curated commonsense knowledge bases with new assertions by formulating the problem as traditional KBC methods used with factual knowledge bases. They devised two neural network models; bilinear and Deep Neural Network; to embed terms and provide scores to arbitrary triples. Both models assumed term embeddings are fixed and learned the best relation representations connecting term pairs. Term embeddings on the other hand are learned from general word embeddings through averaging or applying LSTM on the embeddings of words constituting a terms. To further maximize model accuracy, they trained word embeddings from the original context of terms. Traditionally, KBC methods predict the top k entities that can form a tuple with specified entity and relation $(h, r, ?)$ or $(?, r, t)$. This model however aims to score arbitrary tuples based on their plausibility. Their main goal is to do on-the-fly KBC so that queries can be answered robustly without requiring the precise linguistic forms contained in the knowledge base.

Our model is different from this work on that its is trained on both terms and relations simultaneously. Moreover, we focus on learning terms embedding with semantically salient context that encompass more of terms meaning.

Analogyspace (Speer et al., 2008) is a matrix factorization model designated to facilitate reasoning over commonsense knowledge bases. Analogyspace generate analogical closure of the knowledge base by applying singular value decomposition (SVD) on the knowledge graph matrix. The dimensionality reduction step suppress noisy features and keep the salient aspects of the knowledge. The key idea is that semantic similarity can be determined using linear operations over the resulting vectors.

2.3 Comparison to prior work and its limitations

Manual approaches for commonsense knowledge acquisition relied on labor efforts of knowledge engineers and system experts to formalize and codify CSK assertions. To increase the efficiency of knowledge acquisition, this labor-intensive task was then distributed to volunteers through collaborative platforms such as interactive tools, crowd-sourcing websites, and games with purpose. These manual methods produced highly accurate commonsense assertions that are usually unrecoverable from textual

resources. However, they are highly inefficient, limited in size, and suffer from knowledge gaps.

A shift towards large-scale commonsense knowledge acquisition leveraged on textual resources via pattern matching to discover potential valid CSK assertions. These methods can follow either semi-automated approaches that rely on handcrafted extraction patterns (Pasca, 2014; Clark and Harrison, 2009; Etzioni et al., 2004), or automated approaches that utilize bootstrapping methods for patterns generation and facts extraction (Tandon and De Melo, 2010; Tandon et al., 2011). In general, text-mining based methods are inherently limited to extract explicit or subtly implicit commonsensical assertions. Further, they rely on syntactical extraction patterns which disregard, to a large extent, the semantics associated with the CSK, and thus unable to deal with CSK ambiguity. Despite the high recall and the expanded coverage of these methods, they suffer from low precision and noisy extractions.

Reasoning approaches for CSKA attempt to automatically infer missing knowledge based on pre-existing knowledge. These approaches go beyond the literal extraction of explicit assertions to the elicitation of implicit assertions. Vector space models convert entities and relations of a knowledge base into compact k-dimensional vectors and use these vector representations to predict missing facts. This family of reasoning approaches has the capacity to integrate external sources of information into the representation learning framework. External information can play a key role in understanding and recovering the semantic information associated with abstract concepts. An example is the work of Li et al. (Li et al., 2016) that considered concepts as phrasal terms and learned their representations through word embedding model trained over a textual training set. Representation learning based methods are powerful tools, however, they are highly dependent on the quality of the underlying knowledge. Moreover, they suffer from scalability issues.

In summary, prior work in CSKA are either inefficient and non-scalable manual methods that produce high quality and implicit CSK, or large-scale automatic and semi-automatic methods that produce large collections of rather noisy CSK. Moreover, automatic methods are unable to handle the ambiguity associated with abstract concepts and therefore they can't extract implicit knowledge and can't differentiate between concepts' senses. Table 2.2 compares our approach against related work.

Approach	Sub. Setting	Knowledge type			Performance			Integration	
		K.Type	K.Src	Cov.	Eff.	Prec.	Scal.	Extr.K	Ambiguity
Manual	Curated	CS	Impl.	Low	Low	High	V.Low	No	-
	Collaborative	CS	Impl.	Low	Low	High	Low	No	-
Text Mining	Semi-Automated	F/CS	Expl.	High	Mid	Low	High	No	No
	Automated	F/CS	Expl.	High	High	Low	Mid	No	No
Reasoning	Induction	CS	Impl.	Fill Gaps	High	Low	low	No	No
	Repr. Learning	CS	Impl.	Fill	High	Low	Low	Yes	Yes
Gaps									

Table 2.2: Positioning the dissertation against related work.

K.type: Knowledge type [**CS:** Commonsense; **F:** Factual]; **K.Src:** Knowledge Source [**Impl.** Implicit; **Expl.:** Explicit]; **Cov.:**Coverage; **Eff.:** Efficiency; **Prec.:** Precision; **Scal.:** Scalability; **Extr.K:** Use of External Knowledge; **Ambiguity:** Resolve Ambiguity.

2.4 Applications

Commonsense knowledge can serve wide range of tasks and commercial applications spanning diverse domains like NLP, robotics, and computer vision as well as high-level applications in search engines. We briefly describe some of these application:

- **Expert systems:** Traditional expert systems (ESs) are designed to simulate the judgement and behaviour of a human expert on a particular subject fields, including in financial services, telecommunications, healthcare, customer service, transportation, etc. Typically, an expert system consist of task-specific knowledge base of accumulated human experience and set of rules designed for pre-defined problems and situations. These ESs break down when faced with new situations. To Expand beyond their original scope such that they can better approximate human judgement in new situations, ESs need possessing commonsense knowledge and learning capabilities over this knowledge (McCarthy, 1984; Lenat et al., 1985).
- **NLP:** The important role of commonsense knowledge for natural language processing tasks such as disambiguation and machine translation was discuss by Bar-Hillel (Bar-Hillel, 1960) in as early as 1960. CSK is particularly significant in cases that can't be resolved by simple human-coded rules, rather, requires a actual understanding of real-world knowledge. For example, machine translation, one of the most challenging and unresolved tasks in NLP, needs to go beyond literal word to word mapping which would result on an incorrect or odd translations to meaning mapping which requires a fundamental understanding of the syntax and semantics of source and target languages. Other examples include sense disambiguation (Dahlgren and McDowell, 1986; Curtis et al., 2006; Havasi et al., 2010), textual entailment (Chen and Liu, 2011), sentiment analysis (Cambria et al., 2015a), story understanding and generation (Liu and Singh, 2002; Ong, 2010; Williams, 2017), and handwriting recognition (Wang et al., 2013).
- **Computer vision:** similar to NLP, commonsense has a fundamental role in advancing some essential computer vision task such as image interpretation (Xiao et al., 2010), object detection (Rohrbach et al., 2011), and texttoscene conversion (Coyne and Sproat, 2001)
- **Robotics:** Commonsense reasoning is an intrinsic requirement for autonomous robots working in an uncontrolled environment. Autonomous robots should

be able to understand the world around it and able to interrupt scenes. For instance, a robot that is expected to interpret a scene of a person doing rock climbing should have an understanding of the semantics in the scene. A household robot is expected to guess the desires of a user based in its current beliefs and commands (Kunze et al., 2010; Tenorth et al., 2010).

- Intelligent systems: Search engines or question answering systems such as personal assistants or visual question answering (Antol et al., 2015) can convert a question into some kind of query against a knowledge base to enrich search results with structured information. Moreover, lower error rates in speech recognition powered personal assist systems like Siri, Alexa, and Google Go.

Chapter 3

Models

3.1 Semantically Enhanced KGE Models for CSKA

Reasoning based methods for commonsense knowledge acquisition make rough guesses of valid commonsense assertions based on analogies and tendencies derived from regularities in known commonsense knowledge. By representing a knowledge base as graph consisting of nodes (entities) connected by edges (relations), knowledge graph embedding models learn embeddings of graph entities and relations in low-dimensional continuous vector spaces that preserve graph properties and structural regularities. These embeddings can then be used in downstream tasks such as entity classification, relation extraction, and link prediction. One particular task that we are interested in and that can benefit from these embeddings is knowledge base completion. Knowledge base completion is a follow up step in knowledge acquisition. It is defined as the task of predicting new assertions that are not originally in a knowledge base by filling missing entries of incomplete triples.

Definition 3.1.1: Knowledge Base Completion

Given knowledge assertions represented in form of triples, i.e. (h, r, t) , and scoring function $f_r(h, r, t)$ that score correct triples higher than incorrect triples, knowledge base completion find missing entries $\underline{e}$ of incomplete triples of form $(h, t, ?)$, $(?, r, t)$, or $(h, ?, t)$ such that $\underline{e}$ maximizes the scoring function $f_r(h, t, \underline{e})$, $f_r(\underline{e}, r, t)$, or $f_r(h, \underline{e}, t)$.

A key factor to the performance of these models is the ability of the embeddings to encode as much as possible of structural properties and semantic information of

the knowledge graph. Models for knowledge graph embedding learning fall into two main categories:

1. Models that depend solely on graph structural information.
2. Models that combine structural information with external data resources.

In the latter; let's call them compositional models; the external data resources provide insight into entities' and relations' semantics at both local and global levels. Differences between models lay on the type of external information utilized and the composition methods applied. When dealing with encyclopaedic knowledge in which entities refer to concert world objects, entities semantics are commonly obtained from general textual corpora which serve as a source of diverse contexts in which an entity has appeared. Previous work in this direction utilized entities' description (Zhong et al., 2015; Xie et al., 2016), Wikipedia anchors (Wang et al., 2014a), newspapers (Han et al., 2016), entities' original phrasal form (Li et al., 2016), etc. Some approaches adopted more sophisticated context definitions, such as graph paths (Lin et al., 2015a; Guu et al., 2015; Toutanova et al., 2016) and syntactic parsing of entities mention (Toutanova et al., 2015).

Unlike encyclopaedic knowledge, commonsense knowledge is concerned with abstract concepts that can be manifested in different textual forms in natural texts. In addition, assertions involving abstract concepts are commonly expressed in subtly implicit manner. The abstract and implicit characteristics make traditional compositional knowledge graph embedding models insufficient to capture the structural and semantic regularities in commonsense assertions. To overcome these limitations, we need to promote improvements in knowledge graph embeddings by building semantically focused contextual information that can provide better insight into entities' and relations' semantics, and which will subsequently improve the performance of automatic knowledge acquisition.

Here we present a compositional approach to improve commonsense knowledge graph embeddings with the aim of enriching these knowledge graphs with new assertions. We follow the approach that combines graph structural information with external information. We draw on the idea that importing semantically refined contextual information to commonsense knowledge graph representation learning will result in more focused embeddings (Chen and de Melo, 2015). Having obtained compact vector representations encoding concepts and relations both connectivities and semantics, we can utilize them to perform knowledge reasoning to predict new assertions. Through out this thesis, we use ConceptNet as the commonsense knowledge

base to learn graph and semantic embeddings and to perform knowledge reasoning and acquisition. ConceptNet consist of a big number of concepts connected by a fixed set of 38 relation types. We further use three semantic resources to incorporate in our model.

3.1.1 Problem Formulation

We begin by introducing notation to formally define the problem of *semantically enhanced knowledge graph embedding models for commonsense knowledge acquisition*. A commonsense knowledge base is represented as a graph $\mathcal{G} = \{\mathcal{C}, \mathcal{R}, \mathcal{T}\}$, where $\mathcal{C}$ is the set of concepts, $\mathcal{R}$ is the set of relations, and $\mathcal{T}$ is the set of triples. Each triple represents head and tail concepts connected through a relation, e.g., (*Victory*, *Causes*, *Celebration*) and is denoted as (h, r, t) such that $h, t \in \mathcal{C}$ and $r \in \mathcal{R}$.

Given a set of triples $\mathcal{T}$, our objective is to predict new commonsensical assertions that are not originally in the knowledge base by filling missing entries of incomplete triples of form $(h, r, ?)$, $(?, r, t)$, or $(h, ?, t)$ such that the predicted concept or relation belongs to the existing $\mathcal{C}$ and $\mathcal{R}$, respectively and (h, r, t') , (h', r, t) , $(h, r', t) \notin \mathcal{T}$ where h', t' , and r' are the predicted concepts and relations. To accomplish this, we aim to learn the vector representations of concepts $\mathbf{h}, \mathbf{t}$ and relations $\mathbf{r}$ in $\mathbb{R}^d$ that utilize various information resources and use these vector representations to asses the correctness of a triple through a score function $f_r(h, r, t)$ characterized by the relation r . In the context of knowledge graph representation learning h and t are referred to as entities, therefore, *concept* and *entity* are used interchangeably for the rest of the thesis. Our proposed model is thus of two parts, **(1) Knowledge Representation Model** and **(2) Semantic Representation Model**. The overall architecture of the model is illustrated in figure 3.1.

Definition 1. Knowledge Representation Model: This model learn representations solely from the observed triples using knowledge graph embedding models. KGE models learn low-dimensional vector representations of KG entities and relations such that the learned embeddings maximize a scoring function that measure the plausibility of each individual triple, and collectively, measure the total plausibility of all observed triples in KG. Each concept c and relation r has a knowledge-based vector representations $\mathbf{c}_k$ and $\mathbf{r}_k$ respectively.

Definition 2. Semantic Representation Model: This model learn repre-

Figure 3.1: Model Architecture

sentations from external information resources that encompass some semantics of concepts in the knowledge graph e.g. concept description, concept original phrase form, and many others. In this thesis, each concepts $c \in \mathcal{C}$ has a set of semantic descriptions S_c , such that S_j is the j^{th} class of semantic descriptions and $s_{i,c}$ is the i^{th} semantic description of of concept c . Concepts have separate embedding $\mathbf{c}_{s_i}$ for each semantic description $s_{i,c}$.

3.1.2 Proposed Method

As mentioned above, to enhance the quality of knowledge graph embedding in order to better perform KBC, we propose a knowledge graph representation learning model in which representations are derived from multitude of information resources. At high level, this model can be divided into two main parts. The knowledge-based model captures the inherent structure of the knowledge graph, and the semantic-based model captures the multidimensional aspects of concepts from external semantic resources. Each model has a scoring function $f_r(h, r, t)$ that we aim to learn embeddings that maximize its value. We score triples using energy function $E(h, r, t)$ that have low value for correct triples and high value otherwise. Accordingly, our score function becomes $f_r(h, r, t) = -E_r(h, r, t)$. For each model we want to **maximize** $f_r(h, r, t)$ or, in other words, **minimize** $E_r(h, r, t)$. The two models are learned jointly through

minimizing the following overall energy function:

$$E = E_K + E_S \quad (3.1)$$

where E_K is the energy function of knowledge-based representations, while E_S is the energy function of semantic-based representations. For each semantic description S_j , semantic and knowledge representations are enforced to be compatible with each other as follow:

$$E_{S_j} = E_{S_j S_j} + E_{S_j K} + E_{K S_j}, \quad (3.2)$$

where,

$$E_{S_j S_j} = \|\mathbf{h}_{s_j} + \mathbf{r} - \mathbf{t}_{s_j}\|, \quad (3.3)$$

$$E_{S_j K} = \|\mathbf{h}_{s_j} + \mathbf{r} - \mathbf{t}_k\|, \quad (3.4)$$

$$E_{K S_j} = \|\mathbf{h}_k + \mathbf{r} - \mathbf{t}_{s_j}\|. \quad (3.5)$$

and where E_S can be one or the summation of all E_{S_j} .

The overall energy function will project the two types of concept representations into the same vector space while the relation representation is shared and updated by all energy functions.

3.1.3 Knowledge Representation Model

The knowledge model scores each triple based solely on the internal links, hence capture the local connectivity patterns of the knowledge graph. In this model, a link between two entities is an operation on their vectors. Some prominent models are: TransE that scores a triple through an energy function which consider a relation as a translation from head to tail entity such that $\mathbf{h} + \mathbf{r} \approx \mathbf{t}$, and TransR (Lin et al., 2015b) that extends TransE such that entities and relations are embedded into distinct entity and relation spaces $\mathbb{R}^d$ and $\mathbb{R}^m$, respectively. TransR define projection matrix $\mathbf{M}_r \in \mathbb{R}^{d \times m}$ to obtain relation-specific entities projections $\mathbf{h}_r = \mathbf{h} \mathbf{M}_r$ and $\mathbf{t}_r = \mathbf{t} \mathbf{M}_r$. Triples are then defined as translation between the projected entities representations instead $\mathbf{h}_r + \mathbf{r} \approx \mathbf{t}_r$. Another model is structured embedding that scores a triple via a bilinear score function of form $f_r(h, r, t) = \mathbf{h}^T \mathbf{M}_r \mathbf{t}$. In this work we adopt the basic TransE model, thus knowledge model energy is defined as:

$$E_K = \|\mathbf{h}_k + \mathbf{r} - \mathbf{t}_k\| \quad (3.6)$$

where E_K is expected to have a low value for correct triples and high value otherwise. Numerous KGE models can be used to define E_K (a comprehensive review of these models in (Wang et al., 2017)).

3.1.4 Semantic Representation Model

Much insight can be brought into knowledge graph embeddings through the semantics of concepts and relations between them. Concepts are high level abstractions that can encapsulate diverse meanings and inferences. Large part of retrieving concepts semantics is by deriving a meaningful contexts expressing some of their meanings, and integrating these contexts into the representation leaning model. To accomplish this, we derive our knowledge graph concepts' semantics from three information resources as follows:

3.1.4.1 Textual semantics

Commonsense knowledge bases connect concepts, in the form of words and phrases of natural language, with labelled edges. Knowledge embedding models consider concepts and relations as symbolic elements and recover their structural relatedness and regularities. However, words and phrases as standalone elements carry rich semantic information. Word embeddings, such as word2vec (Mikolov et al., 2013a) and GloVe (Pennington et al., 2014), capture words generic semantic and syntactic information from large corpora through optimizing task-independent objective function that is agnostic to their structural connectivity. Inferences involving commonsense concepts can largely benefit from concept semantic embeddings when injected into the knowledge representation learning process. This is particularly true for concepts with few training instances, in which case, degrading the quality of knowledge-model embeddings. Thus, semantic relatedness between two concepts' phrases can be measured as

$$-\|\mathbf{h}_t + \mathbf{r} - \mathbf{t}_t\|$$

where $\mathbf{h}_t$ and $\mathbf{t}_t$ are the semantic embeddings of the two concepts phrases. One way to obtain $\mathbf{h}_t$ and $\mathbf{t}_t$ is by averaging word vectors of h and t .

When word and entities embeddings are in different spaces, they are not useful for any computation. To address this, the energy function of the textual semantic model is formulated as in 3.2 to enforce both representations to be compatible:

$$E_T = E_{TT} + E_{TK} + E_{KT} \quad (3.7)$$

such that

$$E_T = \|\mathbf{h}_t + \mathbf{r} - \mathbf{t}_t\| + \|\mathbf{h}_t + \mathbf{r} - \mathbf{t}_k\| + \|\mathbf{h}_k + \mathbf{r} - \mathbf{t}_t\| \quad (3.8)$$

The textual semantics model starts by initializing concepts with semantic embeddings then optimize the aforementioned energy function to fine-tune them to be consistent with their knowledge embedding counterpart.

3.1.4.2 Affective Valence

Affective valence is one aspect associated with natural language concepts. Recent models for concept-level sentiment analysis associate concepts with values encoding their affective valence information (Cambria et al., 2015a; ?). These models define a notion of relatedness between concepts according to their semantic and affective valence. AffectiveSpace (Cambria et al., 2015a) is a novel vector space model for concept-level sentiment analysis that allows semantic features associated with concepts to be generalized and, hence, allows concepts to be intuitively clustered according to their semantic and affective relatedness. AffectiveSpace was built by means of random projection to reduce the dimensionality of affective commonsense knowledge. Specifically, the random projection was applied on the matrix representation of AffectNet. AffectNet is an affective commonsense knowledge base built upon ConceptNet, the graph representation of the Open Mind corpus, and WordNet-Affect (Strapparava et al., 2004), an extension of WordNet Domains, including a subset of synsets suitable to represent affective concepts correlated with affective words. This vector model lend itself as powerful framework that can be embedded in potentially any cognitive system dealing with real-world semantics. Thus, we inject these affective vectors into knowledge-based representation learning with the aim of discovering potential assertion between concepts based on their affective relatedness. We define the affective semantic energy function E_A as:

$$E_A = E_{AA} + E_{AK} + E_{KA} \quad (3.9)$$

Where $E_{AA} = \|\mathbf{h}_a + \mathbf{r} - \mathbf{t}_a\|$, $\mathbf{h}_a$ is the affective vector produced by AffectiveSpace, and E_A is expanded analogically to 3.8.

3.1.4.3 Common Knowledge

"You shall know a word by the company it keeps" (Firth, 1957) is a principle that underpinned many text and graph embedding models. For example, word2vec skip-gram model predicts a word from its context, and node embedding models such as

Deepwalk (Perozzi et al., 2014), LINE (Tang et al., 2015) and node2vec (Grover and Leskovec, 2016) learn node embeddings based on their first-order or second-order neighbourhood. Similarly, compositional KGE models link entities and relations with various types of textual context and use them to learn entities and relations embeddings in joint framework. Most of these models inject word embeddings into the process of representation learning of the corresponding entities. Researchers have promoted diverse textual resources as context for entities’ semantic representation learning such as entities’ description (Zhong et al., 2015; Xie et al., 2016), Wikipedia anchors (Wang et al., 2014a), newspapers (Han et al., 2016), entities’ original phrasal form (Li et al., 2016), etc. Some approaches adopted more sophisticated context definition, such as graph paths (Lin et al., 2015a; Guu et al., 2015; Toutanova et al., 2016) and syntactic parsing of entities mention (Toutanova et al., 2015). In the same vein, but for commonsense concepts, Chen and de Melo (Chen and de Melo, 2015) suggested using concept definitions and lists as focused contexts for concept embeddings. Inspired by this work, we propose new semantic context definition that have a potential to provide a boost in concepts embeddings expressiveness.

Since concepts are high level abstractions and given the implicit nature of their mentions, their diverse meanings might be difficult to retrieve from text. One way to recover some of these meanings is through examining instances connected with concepts via hyponym-hypernym relations. These instances carry sub-meanings of their more general superordinates, thus, carry focused semantic inferences.

We propose new semantic context definition that have a potential to provide a boost in concepts embeddings expressiveness. In our model, we aim to recover as much as possible of instances categorized under each concept and integrate their embeddings into our knowledge model. That is, for each concept $c \in \mathcal{C}$, we retrieve a list of instances $I_c = \{I_{c,1}, I_{c,2}, \dots, I_{c,n}\}$, where $I_{c,j}$ is the j^{th} instance of concept c and n is the total number of instances of concept c . These instances are then used to construct a *common-knowledge* embedding $\mathbf{c}_c$. Assuming each instance $I_{c,i}$ has embedding $\mathbf{I}_{c,i}$, the *common-knowledge* embedding of concept c is defined as:

$$\mathbf{c}_c = \frac{1}{n} \sum_{I_{c,i} \in I_c} \mathbf{I}_{c,i} \quad (3.10)$$

the average encoder can be replaced by LSTM or non-linear transformation. The final semantic energy E_C function for this external resources is then:

$$E_C = E_{CC} + E_{CK} + E_{KC}$$

where $E_{CC} = \|\mathbf{h}_c + \mathbf{r} - \mathbf{t}_c\|$, and E_C is expanded analogically to 3.8.

3.2 Sense Disambiguated KGE Models for CSKA

Typically, knowledge graph embedding models represent entities with a single vector per entity, derived from the inherit structure in the knowledge graph (Bordes et al., 2013; Wang et al., 2014b; Lin et al., 2015b; Shi and Weninger, 2017) and from entities semantic and syntactic information in textual resources (Wang et al., 2014a; Toutanova et al., 2015; Xie et al., 2016; Wang and Li, 2016). The structural regularities and semantic meanings captured by these vectors can then be used to perform analogical reasoning, leading to many useful applications, such as probabilistic knowledge acquisition via knowledge base completion or triple scoring (Angeli and Manning, 2013; Li et al., 2016).

In commonsense knowledge bases, concepts are abstract textual terms (words or multi-word phrases) that can have a single meaning (monosemous) or multiple meanings (homonymous or polysemous). For instance, the concept “*program*” appeared in ConceptNet semantic network with different meanings including: 1. a computer program (noun), 2. a radio or television show (noun), and 3. writing a computer program (verb) (figure 3.2 top). Therefore, the structural regularities in the concept “*program*” local connections might be obscured, and the complementary semantic information derived from auxiliary semantic resources of the concept term have the limitation of conflating all concept meanings into a single vector representation.

Thus, a single embedding might be incapable of representing all possible meanings, also called senses, of a concept; a deficiency that would hamper the effectiveness of these embeddings in analogical reasoning and link predication. Therefore, disambiguating concepts’ senses in knowledge base triples would resolve much of the structural irregularities and semantic ambiguity associated with concepts and will shift the embedding paradigm from concept-level representation learning to fine-grained sense-level representation learning, which would eventually improve, knowledge acquisition.

(a) Origianl

(b) Sense Disambiguated

Figure 3.2: Snapshot of a knowledge graph

In this part, we propose sense-aware knowledge graph embedding model for commonsense knowledge acquisition. The model disambiguates concepts in a knowledge base to their senses then embed the sense-disambiguated knowledge base concepts

into low-dimensional vector space that encodes the various senses of concepts with sense-specific embeddings. These embeddings are then used to infer new assertions by means of analogical reasoning. Concepts' senses are induced by analysing textual corpora in which they have appeared. In particular, the textual contexts in which a concept has appeared are clustered into groups denoting the concept's different senses, and the sense of a concept is chosen by determining the sense-cluster with the highest similarity to the current context of the concept. Two steps follow from here: first, concepts in the knowledge base are broken down into their respective senses (Figure 3.2 bottom), and second, sense-specific semantic embeddings for each concept are trained via word embedding model. The original knowledge base is then expanded, with each concept decomposed to instances equal to its senses, then text-enhanced knowledge graph embedding models are trained over the expanded knowledge base, where the sense-specific semantic embeddings learned earlier serve as the auxiliary semantic source for the KGE models in fashion similar to that in 3.2. In the next step, new assertions are predicted using KBC, and triple classification. The model allows for all different context embedding calculation, clustering algorithms, and knowledge graph embedding models.

3.2.1 Problem Formulation

A knowledge graph is denoted as $\mathcal{G} = \{\mathcal{C}, \mathcal{R}, \mathcal{T}\}$, where $\mathcal{C}$ is the set of concepts, $\mathcal{R}$ is the set of relations, and $\mathcal{T}$ is the set of triples (h, r, t) , $h, t \in \mathcal{C}, r \in \mathcal{R}$, and a text corpus is denoted as $\mathcal{D}$. Each concept $c \in \mathcal{C}$ is associated with set of context sentences D_c from the text corpus, and $\mathbf{d}_c$ is the vector representation of the context sentence $d_c \in D_c$. Furthermore, concept's contexts are grouped in Z clusters, with different Z values for different concepts.

Definition 1. Concept-sense cluster: $\pi_z(c) = \{d_1, d_2, \dots, d_n\}, d_i \in D_c, z = \{1, 2, \dots, Z\}$, and $n \leq |D_c|$ are the partitioning of concept's c context sentences D_c into Z clusters.

Definition 2. Sense cluster centroid: $\pi_z(c) = \text{Aggregate}(\mathbf{d}_c), d_c \in \pi_z(c)$ is the aggregation of the vector representations of all contextual sentences in cluster $\pi_z(c)$.

Definition 2. Concept-sense semantic embedding: c_z is the semantic representation of the sense-disambiguated concept c_z , learned by general word embedding model trained over sentences in $\pi_z(c)$.

Given graph $\mathcal{G}$ and corpus $\mathcal{D}$, our objective is to learn the concept-sense clusters

of each concept in $\mathcal{C}$, in order disambiguate the knowledge graph such that $\mathcal{G}'' = \{\mathcal{C}'', \mathcal{R}, \mathcal{T}''\}$, $\mathcal{C}'' = \{\bigcup c_z | c \in \mathcal{C}, z \in [1 : Z]\}$, and $\mathcal{T}''$ is the triple expanded after paring concepts with senses. Our ultimate goal is to perform GKE embedding over $\mathcal{G}''$ and utilize the produced embedding for commonsense knowledge reasoning.

3.2.2 Proposed Model

At high level, the sense-aware knowledge graph embedding model works as following:

1. Induce distinct senses associated with concepts in a commonsense knowledge base 3.2.4.
2. Learn sense-specific semantic embeddings for each sense of each concept 3.2.4.
3. Expand the commonsense knowledge base/graph by breaking down each concept into its senses, where a concept instance in a triple is associated with the most probable sense of its induced senses.
4. Run knowledge graph embeddings models, both stand alone and text-enhanced models, on the expanded knowledge graph, and perform KBC.

3.2.3 Sentence Embedding

Let $d_c = \{w_1, \dots, w_{t-1}, w_{t+1}, \dots, w_l\}$ be the context sentence of the concept c in position t , where the maximum length of the sentence is limited by $l = m$. The embedding of context sentence d_c is defined as the weighted average of its individual words' embeddings:

$$\mathbf{d}_c = \frac{1}{|d_c|} \sum_{w_i \in d_c} u(w_i) \mathbf{w}_i \quad (3.11)$$

where $u(\cdot)$ is the weighting function that captures the importance of word w in the corpus $\mathcal{D}$, and $\mathbf{w}_i$ is its word embedding learned using general word embedding model. Here, we used the tf-idf as the weigh function, and word2vec (Mikolov et al., 2013a; Mikolov et al., 2013c) word embedding that contains 300-dimensional vectors for 3 million words and phrases trained on part of Google News dataset (about 100 billion words).

3.2.4 Context Clustering and Sense Induction

Specifying the optimal number of senses associated with a word is one of the challenges of meaning partitioning (Gale et al., 1992; Schütze, 1998; Erk et al., 2009;

Erk, 2012). There are two main approaches followed in the literature. One approach derives a fixed number of senses for each word from curated sense inventories, such as WordNet (Fellbaum, 1998) that lists all possible meanings a word can take. The second approach rely mainly on inducing word senses by analyzing the contexts in which it occurs. Despite that the first method appears to be more straightforward, it has some limitations: (1) some of the senses in text corpus might not be covered in the sense inventory, and (2) some senses in the sense inventory might not be present in the text corpus. Therefore, we resort to the text driven approach for sense induction. This method examines all the context sentences in which a concept has appeared in and try to group them in clusters corresponding to meanings, based on some clustering criteria.

Typically, clustering algorithms takes the number of clusters as input, implying an assumption of fixed number of senses per concept. However, this assumption is unrealistic generalisation. Mainly, because most of English words have a single meaning (monosemous), while the number of meanings of homonymous and polysemous words can vary greatly. For example, 80% of words in WordNet are monosemous, and less that 5% of words have less than 3 meanings. Taking this into consideration, we learn a varying number of senses per concept via a two stage clustering pipeline. In first stage, we follow the work of (Neelakantan et al., 2015) that apply a non-parametric procedure to induce the number of clusters in online procedure. The number of clusters induced in this stage are then used as input to the second stage that perform spherical k -means and k -means clustering over the same sentences set. The main intuition behind the two step clustering is that the online clustering might produce different clusters depending on order of processing contexts. In the second stage, we perform clustering through multiple iterations and pick the most combat clustering. Below, we describe these clustering algorithms in more details.

Online Non-Parametric Clustering: In this clustering process (see Algorithm 1), a new sense cluster for a concept is created every time the maximum similarity between its current context embedding and all its sense clusters’ centroids is below a threshold.

Consider a concept c and let D_c be the set of context sentences associated with c , such that $\mathbf{d}_c$ is the context embedding for $d_c \in D_c$. Concept c is associated with a global semantic embedding $\mathbf{c}_w$ such that $\mathbf{c}_w$ is the average of concept terms’ word embeddings. Our goal is to divide D_c into Z clusters, such that each cluster corresponds to a concept sense/meaning and the value of Z is learned incrementally.

Algorithm 1 Online Non-Parametric Clustering

Input: D_c (Set of context sentences of a concept) λ (Minimum similarity threshold)**Output:** Z (Number of induced senses) $\Pi = \{\pi_1, \pi_2, \dots, \pi_z | z = \{1, 2, \dots, Z\}\}$ (Clusters Centroids) $\Pi = \{\pi_1, \pi_2, \dots, \pi_z\}$ where $\pi_1 = \{d_1, d_2, \dots, d_n\}, d_i \in D_c$ (Clusters membership)

```
1:  $Z \leftarrow 0$ 
2:  $\Pi \leftarrow \{\}$ 
3:  $\Pi \leftarrow \{\{\}\}$ 
4: for  $d_c \in D_c$  do
5:    $\mathbf{d}_c = WAvg(d_c)$ 
6:    $Max.Sim = \max_{z=1,2,\dots,Z} \{sim(\mathbf{d}_c, \pi_z(c))\}$ 
7:    $z_{max} = \arg \max_{z=1,2,\dots,Z} \{sim(\mathbf{d}_c, \pi_z(c))\}$ 
8:   if  $Max.Sim \geq \lambda$  then
9:      $\Pi \leftarrow \{\Pi \setminus \pi_{z_{max}}\} \cup \{\pi_{z_{max}}, d_c\}$ .
10:    Update  $\pi_{z_{max}}$  centroid
11:   else
12:      $\pi_{Z+1} \leftarrow \{d_c\}$ .
13:      $\Pi \leftarrow \{\Pi \cup \{\pi_{Z+1}\}\}$ .
14:      $\pi_{Z+1} \leftarrow \mathbf{d}_c$ 
15:      $Z \leftarrow Z + 1$ 
16:   end if
17: end for
18: return  $Z, \Pi, \Pi$ 
```

Initially, the number of senses per concept are unspecified, thus, we start with an empty set of sense clusters, then we learn them incrementally as the sentences in D_c are processed sequentially. By taking one sentence embedding $\mathbf{d}_c$ at a time, if there are no sense clusters yet, we place the sentence embedding in a new cluster, otherwise, we calculate the similarity between the sentence embedding and all clusters' centroids. If the maximum similarity is above a predefined threshold λ , where λ is a hyperparameter, then the sentence is added to the sense cluster with the maximum similarity, and the cluster centroid is updated with the new sentence embedding, if non of the clusters have similarity score $\geq \lambda$, then a new cluster is created with the sentence embedding. Let Z be the number of context clusters or the number of senses currently associated with concept c , $\pi(c)$ the current sense of concept c is then determined as:

$$\pi(c) = \begin{cases} \pi_{Z+1}(c), & \text{if } \max_{z=1,2,\dots,Z} \{sim(\mathbf{d}_c, \boldsymbol{\pi}_z(c))\} < \lambda \\ \pi_{z_{max}}(c), & \text{otherwise} \end{cases} \quad (3.12)$$

where $z_{max} = \arg \max_{z=1,2,\dots,Z} \{sim(\mathbf{d}_c, \boldsymbol{\pi}_z(c))\}$, and $sim(.,.)$ is any similarity function that measure two vectors relatedness. We use the cosine similarity function as it gives a better measure of the semantic of word vectors than absolute distance (e.g. euclidean). The cluster centroid $\boldsymbol{\pi}_z$ is the average of the sentence embeddings of contexts sentences which belong to that cluster.

Spherical k -means/ k -means: The number of clusters generated by non-parametric clustering may not accurately partition context sentences into their right senses, rather they are indicative of the number of varying meanings a concept has appeared with. Therefore, after obtaining the clusters from the non-parametric algorithm above, we use the induced number of clusters to initialize spherical k -means over the same context sentences D_c . The main difference between the two clustering algorithms is that k -means use Euclidean distance to calculate the distance between the cluster center and a data instance, while Spherical k -means calculate the angle the new data instance make with the cluster center.

3.2.5 Sense-specific Semantic embeddings

After learning the different scenes associated with a concept, we end up with a corpus in which concept mentions are labelled with their corresponding senses. We then use this corpus to learn semantic embeddings of the sense disambiguated concepts. In

particular, we train word2vec CBOW embedding model over the labelled corpus.

Formally given word sequence $w_1, w_2, \dots, w_T$, and given a window size m such that there are m words to each side of a target word, the CBOW model learn word embedding by maximizing the objective function:

$$\frac{1}{T} \sum_{t=1}^T \log p(w_t \mid \sum_{-m \leq j \leq m, j \neq 0} w_{t+j}) \quad (3.13)$$

3.2.6 Sense-Disambiguated knowledge graph embeddings

Having generated commosense knowledge graph with sense disambiguated concepts, we then learn their embeddings using two knowledge graph embedding models TransE and TransR. TransR (Lin et al., 2015b) propose embedding entities and relations into distinct entity and relation spaces $\mathbb{R}^k$ and $\mathbb{R}^d$ respectively. It then defines projection matrix $\mathbf{M}_r \in \mathbb{R}^{k \times d}$ to obtain relation-specific entities projections $\mathbf{h}_r = \mathbf{h}\mathbf{M}_r$ and $\mathbf{t}_r = \mathbf{t}\mathbf{M}_r$. Triples are defined as translation between the projected entities, with as corresponding score function $f_r(h, r, t) = \|\mathbf{h}_r + \mathbf{r} - \mathbf{t}_r\|_2^2$. We train a semantically enhanced variations of both TransE and TransR in the same way as semantic model 3.1.4, however, with the Sense-specific Semantic embeddings $\mathbf{c}_z$ as input. In summary:

Figure 3.3: Simple illustrations of TransE and TransR (Figures adopted from (Wang et al., 2017))

Chapter 4

Datasets and Experimental Setup

4.1 Semantically Enhanced KGE Models for CSKA

4.1.1 Commonsense Knowledge Graph

We tested our approach on a subset of ConceptNet 5.5. We derived our dataset through the following steps. At first, we extracted the English part of conceptNet. This part contains around 1,803,873 concepts, 38 relations, and 28 million triples. Then, from extracted concepts we kept the ones that have counter parts in our auxiliary semantic resources (discussed below). We ended up with a knowledge base of 30,773 concepts, 38 relations, and 366,202 triples, lets call it here CN30K for simplicity. These triples were then divided into training, validation, and test sets. To make these three sets balanced (i.e. each set has enough training examples for each relation type), we first counted triples associated with each relation type, we then divided them with 60%, 20%, and 20% ratios for train, validation, and test respectively. The statistics of the three datasets are illustrated in table 4.1

The resulted knowledge base is highly skewed with majority of triples connect concepts by generic relations, e.g. 80% of triples are connected via `RelatedTo`, `Synonym`, and `IsA` relations, while relations such as `NotHasProperty`, `CreatedBy`, `InstanceOf`, `ReceivesAction`, `DefinedAs`, `LocatedNear`, `MannerOf`, `NotCapableOf`, and `SymbolOf` made up around 1% of triples. The complete relation distribution is illustrated in table 4.2. Furthermore, not all concepts are well represented, with around 15,254 ($\approx 50\%$) concepts have less than 10 occurrences, 8,625 ($\approx 28\%$) have less than 5 occurrences, and 1,882 ($\approx 6\%$) concepts have 1 occurrence only.

Dataset	#Concepts	#Relations	#Triples
Train	30773	38	240246
Validate	20824	38	63992
Test	20234	38	61964
Total	30773	38	366202

Table 4.1: CN30K dataset statistics

4.1.2 Semantics Embeddings

Word2vec and GloVe are two well-known and effective word embeddings that have complimentary strengths over each other (see 2.1.4). Recently, Speer et al. (Speer et al., 2017) presented a novel word embedding model called *Numberbatch*. This model outperformed word2vec and GloVe in the semantic word similarity task of SemEval 2017¹ in addition to other word relatedness and commonsense stories ending. In fact, *Numberbatch* takes word2vec and GloVe word vectors as input and improves on them by the mean of retrofitting (Faruqui et al., 2014), a method to refine existing word embeddings using relation information from external resource. Since *Numberbatch* adjusted word embeddings to reflect their connectivity in ConceptNet 5.5, they serve as a perfect fit for our semantic embeddings model 3.1.4). However, similar semantic embeddings can be obtained for any knowledge base using the retrofitting procedure. Moreover, *Numberbatch* embeddings can be replaced by any semantic distribution model. We further describe the procedure to build *Numberbatch* with more details:

***Numberbatch*:** is state-of-the-art semantic vectors that is built using an ensemble model that combines two generic word embeddings resources: word2vec and GloVe, and one relational data resource: ConceptNet. The model starts by representing ConceptNet multilingual knowledge graph as sparse, symmetric term-term matrix in which each cell holds the sum of weights of edges connecting the two corresponding concepts. The matrix is then used to define the context of each concept. As opposed to regular text corpus in which the context of a word consist of words surrounding it within some distance, here the context of a concept is defined as all

¹<http://alt.qcri.org/semeval2017/task2/>

Relation	Instances	Percentage	Relation	Instances	Percentage
RelatedTo	207797	56.74382%	Causes	715	0.19525%
Synonym	48125	13.14165%	Desires	620	0.16931%
IsA	36145	9.87024%	HasLastSubevent	582	0.15893%
HasContext	14058	3.83886%	HasFirstSubevent	575	0.15702%
Antonym	8539	2.33177%	NotDesires	529	0.14446%
AtLocation	8504	2.32222%	dbpedia	418	0.11414%
DerivedFrom	8092	2.20971%	HasA	324	0.08848%
SimilarTo	7591	2.07290%	Entails	272	0.07428%
UsedFor	3636	0.99289%	MadeOf	181	0.04943%
EtymoRelatedTo	2469	0.67422%	NotHasProperty	111	0.03031%
HasPrerequisite	2413	0.65893%	CreatedBy	103	0.02813%
FormOf	2350	0.64172%	InstanceOf	93	0.02540%
DistinctFrom	2158	0.58929%	ReceivesAction	86	0.02348%
CapableOf	2132	0.58219%	DefinedAs	29	0.00792%
HasSubevent	2049	0.55953%	LocatedNear	20	0.00546%
PartOf	1898	0.51829%	MannerOf	14	0.00382%
MotivatedByGoal	1517	0.41425%	NotCapableOf	3	0.00082%
HasProperty	1266	0.34571%	SymbolOf	2	0.00055%
CausesDesire	786	0.21464%			

Table 4.2: CN30K relation distribution statistics

other concepts to which it is connected. This new defined context is then used to calculate word (concept) embeddings of ConceptNet. The authors followed the point wise mutual information (PPMI) method devised by Levy et al. (Levy et al., 2015) which considers rows as words and columns as context, to measure the strength of association between words and produce the PPMI of the matrix, after which, truncated SVD was applied to reduce vector dimensions to 300. In the next step, the purely structural embeddings were enhanced to produce higher quality semantic vectors by integrating word embeddings generated from text corpus. The authors combined the PPMI generated vectors with word2vec (Mikolov et al., 2013a) and Glove (Pennington et al., 2014) precompiled word embedding vectors by the means of retrofitting (Faruqui et al., 2014), a method to refine existing word embeddings using relation information from external resource. Given word vectors $\hat{w}_i$ from word embedding model, e.g. word2vec, retrofitting infer new vectors w_i , such that they are close to their original value and close to their neighbours:

$$\sum_i^m \left[\alpha_i \|w_i - \hat{w}_i\| + \sum_{(i,*,j) \in \mathcal{T}} \beta_{ij} \|w_i - w_j\|^2 \right] \quad (4.1)$$

where α and β values control the relative strengths of associations, m is the size of vocabulary, and $(i, *, j)$ are all concept pairs in the knowledge graph connected by arbitrary relation. The authors set the values of β_{ij} to the weights of edges connecting the concepts corresponding to w_i and w_j .

4.1.3 AffectiveSpace

We associated each concept in CN30K with a vector encoding its affective valence. We used AffectiveSpace (Cambria et al., 2015a) vector space model developed for concept-level sentiment analysis. AffectiveSpace was built by means of random projection to reduce the dimensionality of affective commonsense knowledge. Specifically, the random projection was applied on the matrix representation of AffectNet, a commonsense knowledge base built upon ConceptNet and WordNet-Affect (Strapparava et al., 2004), an extension of WordNet Domains, including a subset of synsets suitable to represent affective concepts correlated with affective words.

4.1.4 Common Knowledge

We rely on multiple resources to retrieve instances/subordinates of each concept in the commonsense knowledge base, in addition to these instances'/subordinates' embeddings. At the beginning, we need to retrieve instances associated with each concept in the dataset. Then we compute, or use pre-computed embeddings associated with each of recovered instances. At the end, we aggregate the embeddings of all instances associated with each concept through a compositional function.

4.1.4.1 Instances extractions

A straightforward way to obtain instances of each concept is by inquiring other gigantic knowledge bases such as DBpedia and Freebase for instances associated with concepts via **IsA** relation. For example DBpedia has 1,450 concepts connected by over 24 million **IsA** pairs, while YAGO has 352,297 concepts connected through over 8 million **IsA** pairs. ProBase² (Wu et al., 2012) is a recent probabilistic taxonomy of common knowledge organized as a hierarchy of hyponym-hypernym relations. It

²<https://www.microsoft.com/en-us/research/project/probase/>

consist of 5,401,933 unique concepts and 12,551,613 unique instances harnessed from 1.68 billion web pages and represented as (*Entity*, *IsA*, *Concept*) triples. We consider this knowledge base as a source to obtain concepts subordinates.

For each concept $c \in \text{CN30K}$, we query ProBase to recover a list of its corresponding instances $I_c = \{I_{c,1}, I_{c,2}, \dots, I_{c,n}\}$. Extracting instances from ProBase is a two steps process:

1. **Concept matching:** Given ProBase concepts P , for each concept $c \in \text{CN30K}$, find all ProBase concepts $P_c = \{p_1, p_2, \dots, p_k\}, p_i \in P$, that match c . This breaks down to three sub-steps performed in sequence:
 - (a) Concept normalization and standardization.
 - (b) N-Gram concept matching.
 - (c) Semantic concept matching.
2. **Instance matching:** For each $p_i \in P_c$, find all instances it is connected with and add them to I_c list

1. Concept matching: Different knowledge bases express concepts in different forms. Therefore, it is crucial to have a method to define similarity between concepts' textual expressions in order to match concepts across resources.

a. Concept normalization and standardization: In many cases, ProBase concepts are expressed as natural language sentences, e.g. “*economy wide institutional and policy reform*”, or “*state of the art inspection equipment*”. In numbers, 3,485,470 (65%) out of 5,401,933 probase concepts are ≥ 3 -gram. To handle this, we first run Stanford CoreNLP tool ³ over ProBase concepts to convert them into normalized and standardized form. Table 4.3 list some of ProBase concepts verses their standardized forms as produced by CoreNLP tool.

b. N-Gram concept matching: After converting ProBase concepts to standard form, for each concept in CN30K, we retrieve a list of candidate ProBase concepts P'_c using simple n -gram matching, for $n \leq 4$.

b. Semantic concept matching: We measure the semantic similarity a concept c and all its candidate concepts P'_c using cosine similarity function. Each concept is represented as a vector of the average of its words' embeddings. For

³<https://github.com/SenticNet/concept-parser>

ProBase concepts, we average word embeddings of words in the original concept rather than in the standardized concept. For word embeddings, we pre-trained word2vec word embeddings⁴. Concept with similarity above a threshold α are added to the final set P_c . We set $\alpha = 0.5$.

ProBase Concept	Standardized Concepts
Economy wide institutional and policy reform	wide_institutional, institutional_economy, policy_reform, reform_economy
Estate of the art inspection equipment	equipment, equipment_art, inspection_equipment, state_of_equipment
Cluster management application	management_application, cluster
Fundamental object-oriented mechanism	fundamental_mechanism, object-oriented_mechanism
Typical urban environmental issue	urban_issue, environmental_issue, typical_issue

Table 4.3: ProBase concepts standardized by CoreNLP tool

2. Instance matching: Instance matching is straightforward. For each ProBase concept in P_c , we recover all instances associated with **IsA** relation and add them to the concept’s instances list I_c

4.1.4.2 Instance Embedding

Once we have recovered all instances associated with concepts in CN30K, we want to recover their embeddings. The *common-knowledge* embedding $\mathbf{c}_c$ of a concept is the average of its instance embeddings. Since we target the semantics of these instances, distributional semantics vectors such as word2vec and Glove are logical choices. However, in this work, we use rely on a more specified embeddings called Isacore (Cambria et al., 2014) that are derived directly from Probase and ConceptNet.

IsaCore is a resource of common and commonsense knowledge that is a result of partially blending ProBase and ConceptNet knowledge bases. The transformation from Probase to Isacore is a multistep process. (1) First, a semantic network, termed

⁴<https://code.google.com/archive/p/word2vec/>

ConceptNet Concept	ProBase Concept	Instances
form_of_exercise	advanced form of exercise	jogging, weight, cycling, exercise bike
special_event	corporate and special event	exhibition car, mascot, special pickup for vips, surprise for date, proposal
fun_activity	regular and fun activity	salsa dance, zumba class, pilate, salad making workshop

Table 4.4: Examples of CN30K matches in ProBase instances

Isanette is built out of approximately 40 million Probase IsA triples, and represented as matrix of $4,622,119 \times 2,524,453$ dimensions. (2) Next, the network was cleaned using word similarity and multidimensional scaling (MDS) to solve the problem of noise and multiple concept forms. Specificity, at this step, concepts with high word similarity and which are close enough to each other in the vector space generated from Isanette are merged. Further, concepts and instances with low connectivity are discarded leaving Isanette a strongly connected core. (3) To complete Isanette, it was enriched with complementary hyponym-hypernym commonsense knowledge (that is assertions with IsA relations) from ConceptNet, yielding $500,000 \times 300,000$ matrix whose rows are instances (for example, birthday party and china), whose columns are concepts (for example, special occasion and country), and whose values indicate truth values of assertions. (4) Lastly, Semantic Multidimensional Scaling is performed on the resulting matrix; M ; to build a vector-space representation of the instance-concept relationship matrix.

4.2 Sense Disambiguated KGE Models for CSKA

4.2.1 Dataset and Experimental Setup

In this project as well, we obtain triples ConceptNet 5.5 semantic network (Speer and Havasi, 2012). ConceptNet was primarily derived from the Open Mind Common Sense (OMCS) in addition to other resources. ConceptNet triples that were derived from OMCS retain the original sentences that were entered by volunteers from which they were derived. Our dataset consist of OMCS entries in ConceptNet. This results in 612,640 triples with 350,304 unique concepts connected by 32 relations. From this dataset, we derived two datasets, CN_Freq5 and CN_Freq10, which contain concepts with frequency above or equal 5 and 10, respectively. The statistics of these two datasets are showed in table 4.5.

Dataset	#Triples	#Rel.	#Comp.	1-gram	2-gram	3-gram	>3gram
Full	612640	32	350304	83858 (24%)	161700 (46%)	56987 (16%)	47760 (13.6%)
CN_Freq5	243530	32	30391	20531 (67.5%)	8234 (27%)	1336 (4%)	291 (1%)
CN_Freq10	181072	32	14130	10553 (75%)	2843 (20%)	597 (4%)	138 (1%)

Table 4.5: Statistics of datasets for sense disambiguation model.

1-gram=number of 1-gram concepts, 2-gram= number of 2-gram concepts, etc.

We notice that the majority of concepts have low representation in ConceptNet. In our dataset, 319913 concepts (91% of concepts) have less than 5 instances, and only 14130 (4%) of concepts have 10 or more instances. Another observation is that multi-word concepts have less frequencies than single-word concepts. For example, multi-word concepts constitute only 76% of the full dataset, but they constitute less than 33 % of frequent concepts datasets. As for relations distribution in the resulting datasets, as in previous data set CN30K, it is highly skewed with the generic **RelatedTo** relation constituting a large proportion of the resulting triples (Table 4.6).

Relation	Full	Freq ≥ 5	Freq ≥ 10
RelatedTo	25.68588 %	37.83147 %	44.19733 %
IsA	17.95916 %	16.43041 %	13.20966 %
Synonym	14.24311 %	7.38635 %	4.91461 %
UsedFor	6.73772 %	5.32131 %	5.58396 %
AtLocation	4.38283 %	6.56346 %	7.44289 %
HasSubevent	4.19642 %	3.47760 %	3.57813 %
CapableOf	3.84173 %	1.23886 %	0.88970 %
HasPrerequisite	3.82377 %	3.19098 %	3.34673 %
SimilarTo	3.45961 %	3.82252 %	1.94453 %
Causes	2.78842 %	2.72451 %	2.93087 %
PartOf	1.73968 %	1.94431 %	1.43534 %
MotivatedByGoal	1.60028 %	1.23845 %	1.32323 %
HasProperty	1.47215 %	1.00275 %	1.04488 %
HasContext	1.26191 %	1.49755 %	1.20117 %
ReceivesAction	1.02686 %	0.31495 %	0.24907 %
HasA	0.97088 %	0.50178 %	0.46832 %
Antonym	0.87898 %	1.93200 %	2.33387 %
CausesDesire	0.77843 %	0.73091 %	0.79195 %
HasFirstSubevent	0.55236 %	0.50055 %	0.50145 %
Desires	0.54795 %	0.30304 %	0.29711 %
NotDesires	0.50225 %	0.23036 %	0.21151 %
HasLastSubevent	0.47172 %	0.45661 %	0.47991 %
DefinedAs	0.38325 %	0.02874 %	0.02429 %
DistinctFrom	0.37167 %	0.90296 %	1.13877 %
MadeOf	0.08895 %	0.13427 %	0.14524 %
Entails	0.06578 %	0.13879 %	0.14524 %
NotCapableOf	0.05925 %	0.01888 %	0.01988 %
NotHasProperty	0.05712 %	0.05789 %	0.06627 %
CreatedBy	0.04276 %	0.06405 %	0.06848 %
LocatedNear	0.00799 %	0.01231 %	0.014358 %
SymbolOf	0.00065 %	0.00041 %	0.00055 %
InstanceOf	0.00032 %	0.00082 %	0.00055 %

Table 4.6: Full datasets relations statistics

4.2.2 Context Clustering

Our goal is to recover senses associated with each concept by clustering the contextual information in which the concept has occurred. We use the OMCS sentences in ConceptNet as training corpus of contextual information. In the OMCS sentences, concepts are expressed in regular English text, which were then extracted and normalized by developers to a standard form. For instance, the triples (*do_crossword_puzzle*, *MotivatedByGoal*, *exercises_brain*) was extracted from sentence “You would [[do a crossword puzzle]] because [[it exercises your brain]]”. We merge normalized concepts into their sentence in order to combine concepts with their semantic and syntactic context. After merging concepts and sentences in previous example, the new sentence becomes “You would [[*do_crossword_puzzle*]] because [[*exercises_brain*]]”.

To learn the embedding of a concept’s contextual sentence, say the concept *exercises_brain*, we first remove it from the sentence, then we learn sentence embedding as the weighted average of the rest of the sentence words’ embeddings 3.2.3. For word embeddings, we used Google’s word2vec (Mikolov et al., 2013a; Mikolov et al., 2013c) word embeddings that contain 300-dimensional vectors for 3 million words and phrases trained on part of Google News dataset (about 100 billion words). We set the maximum length of a concept’s sentence to $n = 20$ regardless of the position of the concept in the sentence.

We then cluster context sentences embedding over two stages. In the first stage, we apply the online non-parametric clustering algorithm (NP-CLUS) of (Neelakantan et al., 2015). As in the original paper, we use cosine function to measure the similarity between sentence embeddings and cluster centroids. A value of 0 means no similarity, and a value of 1 means exact similarity. To choose a range of similarity thresholds to test our method, we experimented with low and high λ values. Low λ values such as 0.5 and 0.55 resulted in too few clusters, grouping sentences with different meanings into the same cluster. On the contrary, high values such as 0.85 and 0.9 resulted in too many clusters, creating a separate cluster for each one or two sentences. We thus choose a range the falls in between these two extreme values. In particular we experimented with $\lambda = \{0.6, 0.65, 0.7, 0.75\}$. Then we use the number of clusters generated by (NP-CLUS) as input to k -means and spherical k -means algorithms. We run both k -means and spherical k -means with 15 iteration with different centroid seeds to choose the best clustering. Table 4.7 show the count of sense-disambiguated concepts in both datasets after concepts clustering with different thresholds. Small λ produce few clusters/senses while higher values produce many clusters/senses. We will discuss later the affect of this on the models performance.

Dataset	Original size	λ			
		0.6	0.65	0.7	0.75
CN_Freq5	30391	37501	43113	54783	75396
CN_Freq10	14131	18935	22577	30875	46276

Table 4.7: Count of sense-disambiguated concepts generated by different clustering thresholds

We further compare the performance of NP-CLUS, k -means, and spherical k -means based on the sum of distances of sentences embeddings to centroids of the clusters they belong to. Table 4.8 show these distances. We notice that the spherical k -means have small inner distances compared to the other two clustering methods due to the fact that spherical k -means perform normalization over vectors before calculating their cosine similarity. We also noticed that NP-CLUS has slightly smaller clusters distances than k -means, but much higher than spherical k -means. We thus anticipate that senses generated by NP-CLUS will have better quality than k -means, and spherical k -means perform the best among all.

Dataset	Original size	λ			
		0.6	0.65	0.7	0.75
CN_Freq5	NP-Clus	227639	219627	205102	183747
	Sph k -means	97671	91049	81350	70798
	k -means	235778	221885	200637	176845
CN_Freq10	NP-Clus	227639	219627	205102	183747
	Sph k -means	97671	91049	81350	70798
	k -means	235778	221885	200637	176845

Table 4.8: Cluster Inner Distance for CN_Freq5 and CN_Freq5 datasets

4.2.3 Sense Embeddings

After context clustering and sense disambiguation, we end up with sentences with labelled concepts. For example:

Something you might do while making_better_world is *volunteer_1*
volunteer_2 is used in the context of military

The resulting corpus is the same as the original OMCS corpus we obtained earlier, except that concepts are disambiguated. We then train a word embedding model over the disambiguated corpus. In particular, we use the word2vec CBOW model used to train google’s word embeddings dataset with 50 iteration and window size of 10. Further, we set vectors dimensionality to 100 to avoid the bias that might result from the small training set (< 3 million words). These embeddings can server as the semantic auxiliary information we incorporated in previous model.

Chapter 5

Evaluation and Discussion

We conducted extensive experiments to assess the effectiveness and validity of our proposed models. Both models are tested with two tasks: (1) Knowledge base completion, (2) Triple classification/scoring. In particular, experiments aim to:

1. Evaluate the effectiveness of semantically enhanced KGEs on the overall performance of the two tasks.
2. Assess the viability of the sense disambiguation algorithms and the effectiveness of disambiguating commonsense concepts on KGEs and subsequently on the overall performance of the two tasks.

5.1 Training

To obtain entities and relations embedding, the model aims to maximize the following margin-based objective function that discriminates between correct triples and incorrect triples:

$$\mathcal{L} = \sum_{(h,r,t) \in \mathcal{T}} \sum_{(h',r',t') \in \mathcal{T}'} \max(0, \gamma + f_r(h, r, t) - f_r(h', r', t'))$$

where $f_r(h, r, t)$ can be any of the knowledge graph embedding models described earlier and $\max(.,.)$ returns the maximum of two inputs, γ is the margin hyper-parameter, $\mathcal{T}$ denotes the set of true triples (h, r, t) that belong to $\mathcal{G}$, and $\mathcal{T}'$ denote the set of corrupted triples not in $\mathcal{G}$: $\{\{h', r, t) | h' \in \mathcal{C}, (h', r, t) \notin \mathcal{G}\} \cup \{(h, r, t') | t' \in \mathcal{C}, (h, r, t') \notin \mathcal{G}\} \cup \{(h, r', t) | r' \in \mathcal{R}, (h, r', t) \notin \mathcal{G}\}$. Negative triples are constructed by corrupting elements of correct triples randomly. We adopt stochastic gradient descent (SGD) to minimize the above loss function. After a mini-batch, the gradient is computed and the model parameters are updated. The objective function applies

for both models in Chapter 3.

Generating Negative Triples Corrupted triples are constructed by replacing h, t or r of a golden triple (h, r, t) with randomly sampled concepts $(h', t') \in \mathcal{G}$ and $r' \in \mathcal{R}$. Wang et al. (Wang et al., 2014a) defined two strategies for replacing head and tail entities: “unif” denotes the traditional way of replacing head or tail with equal probability, and “bern” denotes reducing false negative labels by replacing head or tail with different probabilities. In this work we apply the “unif” setting.

5.2 Experiments and Results

5.2.1 Knowledge base Completion

The task of knowledge base completion aims to complete a triple (h, r, t) when one of h, t, r is missing. That is, predict h given $(?, r, t)$, or predict r given $(h, ?, t)$. Instead of only giving one best answer, the score function $f(h, r, t)$ ranks a set of candidate entities and relations from the knowledge graph. The knowledge graph completion task has two sub-tasks: entity prediction and relation prediction. The result of each sub-task is reported separately.

Evaluation Protocol Following Bordes et al. (Bordes et al., 2013), for each test triple (h, r, t) , we replace the head/tail entity by all entities in the knowledge graph and calculate the similarity score f_r on the corrupted triples. Entities are then ranked in an ascending order of similarity scores. The same procedure is performed for relation predication, in which case, relations are ranked. We use two measures as our evaluation metrics: (1) mean rank of correct entities; (2) proportion of valid entities ranked in top 10. A good link predictor should achieve lower mean rank or higher Hits@10. This basic setting of the evaluation is called “Raw” setting, called so because all entities in the knowledge graph are evaluated and ranked. In this case, some of the corrupted triples may end up being valid ones from training or validation sets, and the model will be penalized for ranking corrupted triple higher than test triple. To eliminate this issue, we filter out the corrupted triples that exist in all the train, validation and test datasets, this is the “Filter” setting. This setting allows ranking a corrupted triple that exists in the knowledge graph higher than test triple.

5.2.1.1 Semantically Enhanced KGE Models for CSKA

Vectors update We train our model with two settings. At first, we initialize concepts with the pre-compiled semantic representations described in previous sections. In *Fixed* setting, during training, we fix concepts’ auxiliary semantic representations and only update knowledge-based concept and relation representations. In *variable* setting, we update all representations simultaneously.

Implementation To train model, we use learning rate α for SGD among $\{0.001, 0.005, 0.01\}$, the margin γ among $\{0.25, 0.5, 1, 2\}$, the embedding dimension n among $\{50, 80, 100\}$. We further use a fixed batch size of 5000. The optimal parameters are determined by the validation set. Regarding the strategy of constructing negative labels, we use “unif” to denote the traditional way of replacing head or tail with equal probability, the optimal configurations are: $\alpha = 0.01$, and $\gamma = 2, n = 100$.

Results: We consider TransE as the baseline model and compare the performance of each semantic model with the baseline separately. We then compare the joint model of all semantic contexts with the baseline. TransE+TXT denotes textual semantics, TransE+AFF denotes affective semantics, TransE+CK denotes the common knowledge semantic, and TransE+ALL denotes the joint semantic model.

Under *Fixed* setting, we notice that the textual semantic model TransE+TXT delivers the best performance in concept predication while at the same time shows improvements over the baseline in relation prediction. The other models, however, show extreme discrepancy in performance in both tasks. For example, the TransE+AFF and the TransE+CK models have rather poor results in concept predication while delivering remarkable improvements in relation prediction. These are understandable results, since the textual semantic representations were optimized to encode not only words semantics, but also words structural connectivity in a relational knowledge, therefore, they transfer some of concepts relational similarities to relations representations, hence the stability in performance. However, in case of affective valence and the common knowledge semantic models, their representations do not encode any structural information, therefore the vectors learned by TransE+AFF and TransE+CK target relation prediction exclusively, irrespective to concepts structural connectivity.

Under *Variable* settings, however, TransE+AFF and TransE+CK show better generalization capability with continuing to show prominent results for relation prediction, but this time without deteriorating their effectiveness in concept prediction. In Fact, they show comparable results with TransE baseline in concept

Model	Fixed				Variable			
	Mean Rank		Hits@10(%)		Mean Rank		Hits@10(%)	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TRANSE	2477	2453	19.77%	24.29%	2477	2453	19.77%	24.29%
TRANSE+TXT	1059	1039	22.97%	26.59%	1259	1235	21.18%	26.49%
TRANSE+AFF	3749	3728	10.36%	11.08%	1502	1478	20.56%	25.48%
TRANSE+CK	3113	3093	7.39%	7.95%	1386	1362	20.18%	24.83%
TRANSE+ALL	1654	1634	16.88%	18.78%	1089	1065	21.29%	26.37%

Table 5.1: Concept prediction evaluation results

Model	Fixed				Variable			
	Mean Rank		Hits@10(%)		Mean Rank		Hits@10(%)	
	Raw	Filter	Raw	Filter	Raw	Filter	Raw	Filter
TRANSE	11.86	11.73	30.58%	31.24%	11.86	11.73	30.58%	31.24%
TRANSE+TXT	10.53	10.4	35.33%	36.26%	10.08	9.95	43.68%	44.85%
TRANSE+AFF	3.899	3.784	95.57%	95.74%	4.303	4.179	92.02%	92.44%
TRANSE+CK	8.629	8.488	66.16%	66.98%	2.446	2.333	94.62%	94.91%
TRANSE+ALL	3.625	3.51	93.2%	93.57%	5.093	4.969	90.69%	91.2%

Table 5.2: Relation prediction evaluation results

prediction, while TransE+TXT still shows the same consistent behaviour with improvements over both tasks and shows the best performance in concept prediction. Notably, TransE+CK has the highest improvement over all other models in relation prediction, confirming thereby that gaining insight into concept meanings (from its instance) helps recover structural regularities that are more evident in factual knowledge.

Finally, we remark that TransE+ALL get affected by the least performing models in all settings, however, combining highest performing models is believed to perform better than any.

5.2.1.2 Sense Disambiguated KGE Models for CSKA

Implementation We train two KGE models, TransE and TransR, on the sense-disambiguated commonsense knowledge graphs obtained by expanding two main datasets CN_Freq5 and CN_Freq10. We set embedding dimensions for TransE’s enti-

ties and TransR’s entities and relation matrix to $k = m = 100$. We use learning rate $\alpha = 0.01$ for SGD, and margin γ to 1. We further use a fixed batch size of 5000. To generate negative samples in training, we replace head, tail, and relation with equal probability.

Results: At first, we compare the the performance of TransE and TransR on full CN_Freq5 and CN_Freq10 datasets and on the datasets generated by sense-disambiguating CN_Freq5 and CN_Freq10 with three clustering algorithms: online Non-parametric clustering (NP-Clus), spherical k -means (S k -means), and k -means. We compare the performance of different clustering thresholds λ . Via manual inspection, we found that the result of both *Raw* and *Filter* ranking setting are correlated, therefore, we report *Filter* results only.

Table 5.3 and table 5.4 show the results of concept predication task on all datasets. The results marked as bold indicate the best results for each clustering algorithm among different λ values, while underlined results are the best results achieved with one particular λ across different clustering and KGE models. The results show that, in general, TransE performs better than TransR on concept prediction. Moreover, the results show that with all clustering algorithms, the best results are achieved most of the time with $\lambda = 0.65$ or $\lambda = 0.70$. A possible reason for these results is that low λ mean that different concept senses that occur in contexts that are semantically close to each other will result in grouping these senses together in one cluster. In other words, it requires large differences (low similarity) between different senses’ contexts in order to place them in different clusters, while subtle differences will result on different senses being grouped in one cluster. In such case, KGE models will still learn a single vector representation for multiple meanings, but now on a more sparse knowledge graph. On the other hand, higher λ values mean that small differences between a concept sense contexts will place them in different clusters which will result on producing too many senses (as reflected in table 4.7). Intermediate λ values seems to strike the right balance and create more accurate partitioning of concepts senses. This means that TransE and TransR will be able to better capture the structural regularities for different concept senses.

Moreover, we notice that better results are reported on CN_Freq10 dataset. This is a reasonable result, since the sense partitioning step will increase the sparsity of the knowledge graph which affects the performance of KGE models. However, since CN_Freq10 datasets have more occurrences per concept than CN_Freq5, the sparsity

problem is less evident and CN_Freq10 still provide sufficient training example for each concept sense.

We can see also that spherical k -means produces the best result among all clustering algorithms, and both NP-Clus and spherical k -means produce better results than k -means. This would suggest that the sense clusters produced by the former two are better than the one produced by the latter, given that NP-Clus and spherical k -means use cosine similarity measure, while k -means uses euclidean distance. This makes sense since the similarity between concepts is better measured by the angle between their vectors after being shift to origin, rather than by the absolute distance between their vectors.

Lastly, an interesting observation is that the performance of most clustering-threshold combinations is comparable or worst than the performance of original non disambiguated datasets. For example, the best result of Hits@10 for CN_10 dataset was 27.79% compared to the baseline of 25.48%, with improvement of $\approx 2.3\%$. While, this gives the impression than sense disambiguation is non-effective, the results of the semantically enhanced models (table 5.7 and table 5.8) show more encouraging results.

Table 5.5 and table 5.6 show results of relation predication task on both datasets. The aforementioned bold and underline notion apply here as well. In link predication, the superior performance of TransR compared to TransE became evident. We can notice that TransR is doing much better job in all test cases. Moreover, we observe that the concept sense-disambiguation produce bigger improvement in the relation predication task than that in the concept predication task. This can be attributed to the small size of relations compared to concepts, hence the difference/distance between relations' embeddings is more distinctive. In 5.5, TransE with k -means clustering and $\lambda = 0.65$ achieve Hits@10 = 39.35% with approximately 10% improvement over the baseline. Similarly, TransR with spherical k -means clustering and $\lambda = 0.65$ achieve 7% improvement over the baseline with Hits@10 = 41.76%

Moreover, observations similar to those for concept prediction still hold. In particular, the superior performance of spherical k -means over others, and the best λ values. In most cases $\lambda = 0.65$ and $\lambda = 0.70$ give the best performance, but also other threshold values still give improved results over the baseline. The sense disambiguation means that each concept will be replaced by a set of concept-sense pairs and the connections of the original concept will be splitted among concept-senses, which means increasing graph sparsity and reducing the number of training exam-

ples per concept-sense. Therefore, when there are many senses (i.e. $\lambda = .75$), the quality of KGEs is degraded. On the other hand, after sense disambiguation, relation occurrences counts remain the same, hence, the size of training examples per relation remains sufficient, and the sense-disambiguation bringing more structure into graph. This is reflected by improved performance over the baseline. Here again, the spherical k -means provides the best performance among all, and NP-Clus is better than k -means.

Model	Clustering	Mean Rank				Hits@10(%)			
		0.6	0.65	0.7	0.75	0.6	0.65	0.7	0.75
TRANSE	CN_Freq5	2280				22.48%			
	NP-Clus	2367	2398	2016	2748	19.64%	20.64%	24.87%	14.09%
	S k -means	2350	2130	<u>1794</u>	<u>2143</u>	<u>21.67%</u>	22.76%	26.47%	<u>23.06%</u>
	k -means	2413	2647	2459	2904	17.48%	15.39%	16.47%	12.41%
TransR	CN_Freq5	2435				18.28%			
	NP-Clus	2495	2187	2114	2514	15.4%	19.84%	21.62%	14.72%
	S k -means	<u>2246</u>	<u>1877</u>	2134	2276	19.12%	24.21%	20.88%	19.74%
	k -means	2547	2446	2468	2564	14.35%	19.34%	18.69%	17.54%

Table 5.3: Concept prediction evaluation with different clustering algorithms, Dataset=CN_Freq5

As shown in previous results, spherical k -means produced the best results among different clustering algorithms. Moreover, thresholds $\lambda = 0.65$ and $\lambda = 0.70$ produced the best results at both concept and relation prediction. Therefore, we carry the remaining experiments on the sense-disambiguated datasets generated spherical k -means with $\lambda \in \{0.65, 0.70\}$.

As reflected by results in table 5.3 and table 5.4, concept prediction results on CN_Freq5 and CN_Freq10 datasets seems to be more effective than those on sense-disambiguated datasets. However, as mention in 3.2.5, we learn semantic embeddings for each sense-disambiguated concept by training word embedding model on sentences in its concept-sense clusters. These semantic embeddings encode the specific sense of each concept. Further, they are conceptually similar to the *textual semantics* auxiliary information proposed in model 3. Therefore, we perform semantically

Model	Clustering	Mean Rank				Hits@10(%)			
		0.6	0.65	0.7	0.75	0.6	0.65	0.7	0.75
TRANSE	CN_Freq10	1630				25.48%			
	NP-Clus	1683	1627	1682	<u>1715</u>	24.12%	25.74%	23.63%	22.67%
	S <i>k</i> -means	1541	<u>1584</u>	<u>1687</u>	1733	27.79%	<u>26.21%</u>	<u>24.82%</u>	23.06%
	<i>k</i> -means	1702	1734	1825	1812	21.03%	21.59%	20.73%	21.25%
TransR	CN_Freq10	1866				23.28%			
	NP-Clus	1894	1870	1830	1853	21.9%	22.84%	24.62%	<u>23.82%</u>
	S <i>k</i> -means	1872	1820	1884	1899	22.12%	24.85%	23.68%	22.44%
	<i>k</i> -means	1885	1829	1868	1896	22.67%	25.34%	23.31%	22.54%

Table 5.4: Concept prediction evaluation with different clustering methods, Dataset=CN_Freq10

Model	Clustering	Mean Rank				Hits@10			
		0.6	0.65	0.7	0.75	0.6	0.65	0.7	0.75
TRANSE	CN_Freq5	15.23				29.85%			
	NP-Clus	15.36	14.85	12.51	16.73	28.12%	31.12%	33.75%	24.65%
	S <i>k</i> -means	14.78	12.43	11.85	<u>13.54</u>	32.56%	34.65%	38.82%	28.87%
	<i>k</i> -means	14.54	11.75	12.76	18.08	31.84%	39.25%	34.47%	25.41%
TransR	CN_Freq5	12.26				34.54%			
	NP-Clus	11.12	10.93	9.47	17.8	36.62%	37.15%	41.76%	27.73%
	S <i>k</i> -means	<u>10.81</u>	9.45	9.73	15.4	<u>39.12%</u>	41.76%	39.88%	<u>32.43%</u>
	<i>k</i> -means	12.7	11.65	11.98	19.21	31.26%	36.65%	39.09%	23.64%

Table 5.5: Relation prediction evaluation with different clustering algorithms, Dataset=CN_Freq5

enhanced knowledge graph embedding using the sense semantic embedding as the *textual semantics* resource. We train TransE and TransR models by updating both knowledge-base and semantic embedding simultaneously (i.e. *Variable* setting). The semantically enhanced models are denoted TransE+S and TransR+S. For concepts in CN_Freq5 and CN_Freq10, we learn the semantic embeddings of concepts from all of their occurrences in the corpus.

Table 5.7 and table 5.8 Show the results for the semantically enhanced and sense-disambiguated knowledge graph embeddings. The results show that the sense semantic embeddings do indeed improve the performance of both TransE and TransR. Moreover, we observe that these embeddings bring more improvement to TransE model than to TransR. We can also see that the improvement that the semantic embedding bring to the sense disambiguated models is bigger than that to the baseline datasets. For example, the *Mean Rank* for $\lambda = 0.70$ (dataset generated by clustering with $\lambda = 0.70$) in table 5.7 decreased from 1794 to 1377 and the Hits@10 increase from 26.47% to 35.41%. This is bigger improvement than this for CN_Freq5.

Similarly, the results of relation prediction in table 5.9 and table 5.10 show similar improvements, but here again, with the TransR model instead. From these two tables, we can observe the remarkable improvements in the results of the semantically enhanced TransR model. For example, Hits@10 improvements ranged from 47% over the baseline for CN_Freq10 dataset to 58% over the baseline for CN_Freq5 dataset with k -means clustering and $\lambda = 0.65$. This dramatic jump in performance can be attributed to the fact that there are limited number of relations (32) compared to tens of thousands of concepts, and the semantic enhancements of concepts' representations that encode specific senses further narrow down the candidate relations that can connect sense-disambiguated concepts.

5.2.2 Triple Classification

Triples classification aims to judge whether a given triple (h, r, t) is correct or not, which is a binary classification task. This task was previously explored by (Socher et al., 2013b) (Wang et al., 2014b) to evaluate their embedding models.

Evaluation Protocol Naturally, a classification task needs samples with positive and negative labels in order to learn a discriminative classification model. Thus, we construct a negative samples for our training set as follows: for each golden triple

Model	Clustering	Mean Rank				Hits@10			
		0.6	0.65	0.7	0.75	0.6	0.65	0.7	0.75
TRANSE	CN_Freq10	13.46				37.48%			
	NP-Clus	13.54	12.10	13.21	15.40	36.64%	38.41%	35.75%	36.31%
	S <i>k</i> -means	12.54	10.19	13.87	14.25	38.23%	42.76%	34.64%	39.51%
	<i>k</i> -means	13.16	10.86	12.32	16.65	32.68%	40.32%	37.47%	36.41%
TransR	CN_Freq10	11.82				38.28%			
	NP-Clus	11.34	9.43	<u>8.54</u>	<u>9.69</u>	29.41%	44.17%	<u>46.62%</u>	<u>43.72%</u>
	S <i>k</i> -means	<u>10.45</u>	<u>8.72</u>	9.81	12.12	<u>39.78%</u>	<u>43.77%</u>	45.11%	37.32%
	<i>k</i> -means	13.65	11.86	10.81	12.94	35.41%	35.91	37.76%	33.17%

Table 5.6: Relation prediction evaluation with different clustering algorithms, Dataset=CN_Freq10

Model	CN_Freq5		$\lambda = 0.65$		$\lambda = 0.70$	
	MR	H@10(%)	MR	H@10(%)	MR	H@10(%)
TRANSE	2280	22.48%	2130	22.76%	1794	26.47%
TRANSE+S	1989	24.73%	1974	29.40%	1377	35.41%
TRANSR	2435	18.28%	1877	24.21%	2134	20.88%
TRANSR+S	2218	21.47%	1690	26.17%	2007	22.13%

Table 5.7: Concept Prediction with semantic vectors, Dataset=CN_Freq5, MR=Mean Rank, H@10=Hits@10

Model	CN_Freq10		$\lambda = 0.65$		$\lambda = 0.70$	
	MR	H@10(%)	MR	H@10(%)	MR	H@10(%)
TRANSE	1630	25.48%	1584	26.21%	1687	24.82%
TRANSE+S	1421	26.11%	1173	29.74%	1372	27.91%
TRANSR	1866	23.26%	1820	24.85%	1884	23.68%
TRANSR+S	1891	22.15%	1567	26.35%	1627	24.90%

Table 5.8: Concept Prediction with semantic vectors, Dataset= CN_Freq10

we generate three negative triple by randomly switching one of h, r, t at a time with h', r', t' , such that $(h', r', t') \in \mathcal{C}$, and $((h', r, t), (h, r, t'), (h, r', t)) \notin \mathcal{G}$.

The classification decision rule is as follows: for a given triple (h, r, t) , if its score is larger than relation-specific threshold δ_r , it will be classified as positive, otherwise as negative. δ_r is obtained by maximizing the classification accuracies on the valid set, and the results are reported the on test dataset.

Implementation In this experiment, we optimize the objective with stochastic gradient descent (SGD). We apply the same parameter settings as in entity predication task.

5.2.2.1 Semantically Enhanced KGE Models for CSKA

We experiment with CN30K dataset. After generating negative triple, we end up with 247,856 test triples out of which 61,964 are correct triples and 185,892 are corrupted triples and 255,968 validation triples out of which 63,992 are correct triples and 191,976 are corrupted triples.

Result. We measure our models ability to discriminate between golden and corrupted triples. From table 5.11, we can see that in both *Fixed* and *Variable* settings, the TransE+CK semantic model have the highest classification accuracy. We also observe that TransE+AFF have surprisingly better performance than TransE+TXT, and in *Variable* scenario, outperform the basesline. These results are strong indication of effectiveness of semantic models in equipping concepts with discriminative features. Hence resolving part of existing ambiguity and commonsense reasoning in an effective manner.

Model	Accuracy	
	Fixed	Variable
TRANSE	88.73	88.61
TRANSE+TXT	83.66	88.75
TRANSE+AFF	87.85	90.41
TRANSE+CK	92.94	91.72
TRANSE+ALL	90.23	89.59

Table 5.11: Triple classification accuracy for CN30K

5.2.2.2 Sense Disambiguated KGE Models for CSKA

As with previous task, we generate three negative triples for golden triple. We experiment with dataset generated by spherical k -means with different threshold values, as it was shown in previous experiments that it provided the best performance among the rest.

Result. The result follow the same scenario of the previous model. As triple classification is more related to relation scoring rather than concept scoring, TransR and TransR+S outperform all TransE and TransE+S. We can further see the sense semantic embeddings improved the performance of the base line model. In both CN_Freq5 and CN_Freq10 dataset, TransE and TransE+S delivered the best performance with $\lambda = 0.70$, while TransR and TransR+S delivered the best performance with $\lambda = 0.65$

Model	CN_Freq5		$\lambda = 0.65$		$\lambda = 0.70$	
	MR	H@10(%)	MR	H@10(%)	MR	H@10(%)
TRANSE	15.32	29.85%	12.43	34.65%	11.85	38.82%
TRANSE+S	10.76	44.85%	8.12	49.6%	6.78	66.74%
TRANSR	12.26	34.54%	9.54	41.76%	9.73	39.88%
TRANSR+S	4.46	79.4%	3.32	91.68%	4.308	89.85%

Table 5.9: Relation Prediction with semantic vectors, Dataset= CN_Freq5

Model	CN_Freq10		$\lambda = 0.65$		$\lambda = 0.70$	
	MR	H@10(%)	MR	H@10(%)	MR	H@10(%)
TRANSE	13.46	37.48	10.19	42.76	13.87	34.64
TRANSE+S	9.21	56.53	6.21	68.1	7.4	59.11
TRANSR	11.82	45.28	8.72	43.77	9.81	45.11
TRANSR+S	4.41	84.62%	3.01	86.94%	5.126	82.60%

Table 5.10: Relation Prediction with semantic vectors, Dataset= CN_Freq10

Model	Accuracy				
	N_Freq5	$\lambda = 0.60$	$\lambda = 0.65$	$\lambda = 0.70$	$\lambda = 0.75$
TRANSE	82.35%	80.62%	81.76%	83.26%	82.51%
TRANSE+S	84.10%	79.21%	83.27%	86.95%	85.61%
TRANSR	88.49%	87.73%	92.59%	91.66%	91.54%
TRANSR+S	92.12%	89.94%	95.46%	93.66%	94.54%

Table 5.12: Triple classification Accuracy on CN_Freq5

Model	Accuracy				
	N_Freq10	$\lambda = 0.60$	$\lambda = 0.65$	$\lambda = 0.70$	$\lambda = 0.75$
TRANSE	88.11%	86.91%	89.78%	93.21%	83.44%
TRANSE+S	92.46%	91.21%	94.27%	95.48%	84.84%
TRANSR	91.38%	88.79%	91.22%	91.56%	87.54%
TRANSR+S	95.12%	93.16%	96.06%	95.83%	89.64%

Table 5.13: Triple classification Accuracy on CN_Freq10

Chapter 6

Conclusion

6.1 Conclusion

We investigate improved knowledge graph embedding models aiming to improve automatic commonsense knowledge acquisition. In particular, we proposed two enhancement that resolve part of the ambiguity associated with commonsense concepts. In the first enhancement, we consider models that perform joint representation learning from structural and semantic resources. We derive a set of semantically salient contexts that cover syntactic, semantic, affective and taxonomical aspects of concepts. A compositional approach combines the knowledge graph structural information with the refined semantic context into a unified knowledge graph representation learning framework. In the second enhancement, we disambiguate concept senses by analysing their context in text corpus. We further learn sense semantic embeddings for each concepts from its context. We train compositional knowledge graph embedding models over the sense-disambiguated knowledge graphs. Empirical results show that some of the semantic information are indeed effective and have the potential to further improve commonsense knowledge acquisition task. Moreover, results show that disambiguating concepts' senses help knowledge graph embedding models to better capture distinctive semantic and structural feature of each concept, which is reflected positively on the knowledge acquisition tasks.

6.2 Future Work

Future work includes employing different knowledge graph embedding models, using LSTM or non-linear transformation to combine the semantic information before incorporating them into the knowledge model, or adding new semantic resources.

Chapter 7

Appendix A

7.1 List Of publications

Working on this thesis produced the following two publications:

- Alhussien, I., Cambria, E., and NengSheng, Z. (2018). *Semantically Enhanced Models for Commonsense Knowledge Acquisition*. Data Mining Workshops (ICDMW), 2018 IEEE International Conference on. IEEE, 2018.
- Alhussien, I., Cambria, E., and NengSheng, Z. *Context Representation Learning for Multi-prototype Knowledge Graph Embedding*. (In-Print, To be submitted to Journal of Information Processing and Management).

Chapter 8

Appendix B

8.1 Abbreviation

AI	Artificial Intelligence
CSK	Commonsense Knowledge
CSKB	Commonsense Knowledge Base
CSKA	Commonsense Knowledge Acquisition
KB	Knowledge Base
KBC	Knowledge Base Completion
KGE	Knowledge Graph Embedding
OMCS	Open Mind Common Sense
LSTM	Long-Short Term Memory

Bibliography

- Akbik, A. and Löser, A. (2012). Kraken: N-ary facts in open information extraction. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pages 52–56. Association for Computational Linguistics.
- Akbik, A. and Michael, T. (2014). The weltmodell: A data-driven commonsense knowledge base. In *LREC*, volume 2, page 5.
- Anderson, M. L., Gooma, W., Grant, J., and Perlis, D. (2013). An approach to human-level commonsense reasoning. In *Paraconsistency: Logic and Applications*, pages 201–222. Springer.
- Angeli, G. and Manning, C. D. (2013). Philosophers are mortal: Inferring the truth of unseen facts. In *CoNLL*, pages 133–142.
- Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., and Parikh, D. (2015). Vqa: Visual question answering. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2425–2433.
- Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., and Etzioni, O. (2007). Open information extraction from the web. In *IJCAI*, volume 7, pages 2670–2676.
- Bar-Hillel, Y. (1960). The present status of automatic translation of languages. In *Advances in computers*, volume 1, pages 91–163. Elsevier.
- Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM.
- Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013).

- Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.
- Bordes, A., Weston, J., Collobert, R., and Bengio, Y. (2011). Learning structured embeddings of knowledge bases. In *Conference on artificial intelligence*, number EPFL-CONF-192344.
- Cambria, E., Fu, J., Bisio, F., and Poria, S. (2015a). Affectivespace 2: Enabling affective intuition for concept-level sentiment analysis. In *AAAI*, pages 508–514.
- Cambria, E., Livingstone, A., and Hussain, A. (2012a). The hourglass of emotions. In *Cognitive behavioural systems*, pages 144–157. Springer.
- Cambria, E., Rajagopal, D., Kwok, K., and Sepulveda, J. (2015b). Gecka: game engine for commonsense knowledge acquisition. In *The Twenty-Eighth International Flairs Conference*.
- Cambria, E., Song, Y., Wang, H., and Howard, N. (2014). Semantic multidimensional scaling for open-domain sentiment analysis. *IEEE Intelligent Systems*, 29(2):44–51.
- Cambria, E., Xia, Y., and Hussain, A. (2012b). Affective common sense knowledge acquisition for sentiment analysis. In *LREC*, pages 3580–3585.
- Chen, J. and de Melo, G. (2015). Semantic information extraction for improved word embeddings. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 168–175.
- Chen, J. and Liu, J. (2011). Combining conceptnet and wordnet for word sense disambiguation. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 686–694.
- Chen, J., Tandon, N., and de Melo, G. (2015). Neural word representations from large-scale commonsense knowledge. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2015 IEEE/WIC/ACM International Conference on*, volume 1, pages 225–228. IEEE.
- Chen, J., Tandon, N., Hariman, C. D., and de Melo, G. (2016). Webbrain: Joint neural learning of large-scale commonsense knowledge. In *International Semantic Web Conference*, pages 102–118. Springer.

- Chklovski, T. (2003). Learner: a system for acquiring commonsense knowledge by analogy. In *Proceedings of the 2nd international conference on Knowledge capture*, pages 4–12. ACM.
- Clark, P. and Harrison, P. (2009). Large-scale extraction and use of knowledge from text. In *Proceedings of the fifth international conference on Knowledge capture*, pages 153–160. ACM.
- Coyne, B. and Sproat, R. (2001). Wordseye: an automatic text-to-scene conversion system. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 487–496. ACM.
- Curtis, J., Cabral, J., and Baxter, D. (2006). On the application of the cyc ontology to word sense disambiguation. In *FLAIRS Conference*, pages 652–657.
- Dahlgren, K. and McDowell, J. P. (1986). Using commonsense knowledge to disambiguate prepositional phrase modifiers. In *AAAI*, pages 589–593.
- Dreifus, C. (1998). Got stuck for a moment: an interview with marvin minsky. *International Herald Tribune (August 1998)*.
- Ehrlinger, L. and Wöß, W. (2016). Towards a definition of knowledge graphs. In *SEMANTiCS (Posters, Demos, SuCCESS)*.
- Erk, K. (2012). Vector space models of word meaning and phrase meaning: A survey. *Language and Linguistics Compass*, 6(10):635–653.
- Erk, K., McCarthy, D., and Gaylord, N. (2009). Investigations on word senses and word usages. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 1-Volume 1*, pages 10–18. Association for Computational Linguistics.
- Eslick, I. S. (2006). *Searching for commonsense*. PhD thesis, Massachusetts Institute of Technology.
- Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2004). Web-scale information extraction in knowitall:(preliminary results). In *Proceedings of the 13th international conference on World Wide Web*, pages 100–110. ACM.

- Etzioni, O., Fader, A., Christensen, J., Soderland, S., and Mausam, M. (2011). Open information extraction: The second generation. In *IJCAI*, volume 11, pages 3–10.
- Fader, A., Soderland, S., and Etzioni, O. (2011). Identifying relations for open information extraction. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1535–1545. Association for Computational Linguistics.
- Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E., and Smith, N. A. (2014). Retrofitting word vectors to semantic lexicons. *arXiv preprint arXiv:1411.4166*.
- Fellbaum, C. (1998). *WordNet*. Wiley Online Library.
- Firth, J. R. (1957). A synopsis of linguistic theory, 1930-1955. *Studies in linguistic analysis*.
- Gale, W. A., Church, K. W., and Yarowsky, D. (1992). A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26(5-6):415–439.
- Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864. ACM.
- Guu, K., Miller, J., and Liang, P. (2015). Traversing knowledge graphs in vector space. *arXiv preprint arXiv:1506.01094*.
- Han, X., Liu, Z., and Sun, M. (2016). Joint representation learning of text and knowledge for knowledge graph completion. *arXiv preprint arXiv:1611.04125*.
- Havasi, C., Speer, R., and Pustejovsky, J. (2010). Coarse word-sense disambiguation using common sense. In *AAAI Fall Symposium: Commonsense Knowledge*.
- Herdagdelen, A. and Baroni, M. (2010). The concept game: Better commonsense knowledge extraction by combining text mining and a game with a purpose. In *AAAI Fall Symposium: Commonsense Knowledge*.
- Hinton, G. E., McClelland, J. L., Rumelhart, D. E., et al. (1986). Distributed representations. *Parallel distributed processing: Explorations in the microstructure of cognition*, 1(3):77–109.
- Howe, J. (2006). Crowdsourcing: A definition.

- Kunze, L., Tenorth, M., and Beetz, M. (2010). Putting peoples common sense into knowledge bases of household robots. In *Annual Conference on Artificial Intelligence*, pages 151–159. Springer.
- Kuo, Y.-l., Lee, J.-C., Chiang, K.-y., Wang, R., Shen, E., Chan, C.-w., and Hsu, J. Y.-j. (2009). Community-based game design: experiments on social games for commonsense data collection. In *Proceedings of the acm sigkdd workshop on human computation*, pages 15–22. ACM.
- Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., Van Kleef, P., Auer, S., et al. (2015). Dbpedia—a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195.
- Lenat, D. B. (1995). Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38.
- Lenat, D. B. and Guha, R. V. (1989). Building large knowledge-based systems; representation and inference in the cyc project.
- Lenat, D. B., Prakash, M., and Shepherd, M. (1985). Cyc: Using common sense knowledge to overcome brittleness and knowledge acquisition bottlenecks. *AI magazine*, 6(4):65.
- Levy, O., Goldberg, Y., and Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225.
- Li, X., Taheri, A., Tu, L., and Gimpel, K. (2016). Commonsense knowledge base completion. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1445–1455.
- Lieberman, H., Smith, D., and Teeters, A. (2007). Common consensus: a web-based game for collecting commonsense goals. In *ACM Workshop on Common Sense for Intelligent Interfaces*.
- Lin, Y., Liu, Z., Luan, H., Sun, M., Rao, S., and Liu, S. (2015a). Modeling relation paths for representation learning of knowledge bases. *arXiv preprint arXiv:1506.00379*.

- Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015b). Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, pages 2181–2187.
- Liu, H. and Singh, P. (2002). Makebelieve: Using commonsense knowledge to generate stories. In *AAAI/IAAI*, pages 957–958.
- Liu, H. and Singh, P. (2004). Conceptneta practical commonsense reasoning tool-kit. *BT technology journal*, 22(4):211–226.
- Manning, C. D., Raghavan, P., Schütze, H., et al. (2008). *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge.
- McCarthy, J. (1984). Some expert systems need common sense. *Annals of the New York Academy of Sciences*, 426(1):129–137.
- Melamud, O., Goldberger, J., and Dagan, I. (2016). context2vec: Learning generic context embedding with bidirectional lstm. In *Proceedings of CONLL*.
- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Mikolov, T., Yih, W.-t., and Zweig, G. (2013c). Linguistic regularities in continuous space word representations. In *hlt-Naacl*, volume 13, pages 746–751.
- Miller, G. A. (1995). Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.
- Mueller, E. T. (1998). *Natural language processing with Thought Treasure*. Signiform New York.
- Neelakantan, A., Shankar, J., Passos, A., and McCallum, A. (2015). Efficient non-parametric estimation of multiple embeddings per word in vector space. *arXiv preprint arXiv:1504.06654*.
- Niles, I. and Pease, A. (2001). Towards a standard upper ontology. In *Proceedings of the international conference on Formal Ontology in Information Systems-Volume 2001*, pages 2–9. ACM.
- Ong, E. C. (2010). A commonsense knowledge base for generating children’s stories. In *AAAI Fall Symposium: Commonsense Knowledge*.

- Panton, K., Miraglia, P., Salay, N., Kahlert, R. C., Baxter, D., and Reagan, R. (2002). Knowledge formation and dialogue using the kraken toolset. In *AAAI/IAAI*, pages 900–905.
- Pasca, M. (2014). Queries as a source of lexicalized commonsense knowledge. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1081–1091.
- Paulheim, H. (2017). Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic web*, 8(3):489–508.
- Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543.
- Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710. ACM.
- Rohrbach, M., Stark, M., and Schiele, B. (2011). Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1641–1648. IEEE.
- Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533.
- Schubert, L. (2002). Can we derive general world knowledge from texts? In *Proceedings of the second international conference on Human Language Technology Research*, pages 94–97. Morgan Kaufmann Publishers Inc.
- Schütze, H. (1998). Automatic word sense discrimination. *Computational linguistics*, 24(1):97–123.
- Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, 34(1):1–47.
- Shi, B. and Weninger, T. (2017). Proje: Embedding projection for knowledge graph completion. In *AAAI*, volume 17, pages 1236–1242.
- Singh, P., Lin, T., Mueller, E. T., Lim, G., Perkins, T., and Zhu, W. L. (2002). Open mind common sense: Knowledge acquisition from the general public. In *OTM Confederated International Conferences” On the Move to Meaningful Internet Systems”*, pages 1223–1237. Springer.

- Socher, R., Bauer, J., Manning, C. D., et al. (2013a). Parsing with compositional vector grammars. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 455–465.
- Socher, R., Chen, D., Manning, C. D., and Ng, A. (2013b). Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pages 926–934.
- Speer, R. (2007). Open mind commons: An inquisitive approach to learning common sense. In *Workshop on common sense and intelligent user interfaces*. sn.
- Speer, R., Chin, J., and Havasi, C. (2017). Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, pages 4444–4451.
- Speer, R. and Havasi, C. (2012). Representing general relational knowledge in conceptnet 5. In *LREC*, pages 3679–3686.
- Speer, R., Havasi, C., and Lieberman, H. (2008). Analogyspace: Reducing the dimensionality of common sense knowledge. In *AAAI*, volume 8, pages 548–553.
- Strapparava, C., Valitutti, A., et al. (2004). Wordnet affect: an affective extension of wordnet. In *Lrec*, volume 4, pages 1083–1086. Citeseer.
- Tandon, N. and De Melo, G. (2010). Information extraction from web-scale n-gram data. In *Web N-gram Workshop*, volume 7.
- Tandon, N., de Melo, G., De, A., and Weikum, G. (2015). Knowlywood: Mining activity knowledge from hollywood narratives. In *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*, pages 223–232. ACM.
- Tandon, N., de Melo, G., Suchanek, F., and Weikum, G. (2014). Webchild: Harvesting and organizing commonsense knowledge from the web. In *Proceedings of the 7th ACM international conference on Web search and data mining*, pages 523–532. ACM.
- Tandon, N., De Melo, G., and Weikum, G. (2011). Deriving a web-scale common sense fact database. In *AAAI*.
- Tandon, N., de Melo, G., and Weikum, G. (2017). Webchild 2.0: Fine-grained commonsense knowledge distillation. *ACL 2017*, page 115.

- Tandon, N., Hariman, C., Urbani, J., Rohrbach, A., Rohrbach, M., and Weikum, G. (2016). Commonsense in parts: Mining part-whole relations from the web and image tags. In *AAAI*, pages 243–250.
- Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. (2015). Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1067–1077. International World Wide Web Conferences Steering Committee.
- Tellex, S., Katz, B., Lin, J., Fernandes, A., and Marton, G. (2003). Quantitative evaluation of passage retrieval algorithms for question answering. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 41–47. ACM.
- Tenorth, M., Kunze, L., Jain, D., and Beetz, M. (2010). Knowrob-map-knowledge-linked semantic object maps. In *Humanoid Robots (Humanoids), 2010 10th IEEE-RAS International Conference on*, pages 430–435. IEEE.
- Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P., and Gamon, M. (2015). Representing text for joint embedding of text and knowledge bases. In *EMNLP*, volume 15, pages 1499–1509. Citeseer.
- Toutanova, K., Lin, V., Yih, W.-t., Poon, H., and Quirk, C. (2016). Compositional learning of embeddings for relation paths in knowledge base and text. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1434–1444.
- Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 384–394. Association for Computational Linguistics.
- von Ahn, L. (2006). Games with a purpose. *Computer*, 39(6):92–94.
- Von Ahn, L., Kedia, M., and Blum, M. (2006). Verbosity: a game for collecting common-sense facts. In *Proceedings of the SIGCHI conference on Human Factors in computing systems*, pages 75–78. ACM.
- Wang, Q., Mao, Z., Wang, B., and Guo, L. (2017). Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743.

- Wang, Q.-F., Cambria, E., Liu, C.-L., and Hussain, A. (2013). Common sense knowledge for handwritten chinese text recognition. *Cognitive Computation*, 5(2):234–242.
- Wang, Z. and Li, J. (2016). Text-enhanced representation learning for knowledge graph. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI*, pages 1293–1299.
- Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014a). Knowledge graph and text jointly embedding. In *EMNLP*, volume 14, pages 1591–1601. Citeseer.
- Wang, Z., Zhang, J., Feng, J., and Chen, Z. (2014b). Knowledge graph embedding by translating on hyperplanes. In *AAAI*, volume 14, pages 1112–1119.
- Williams, B. M. (2017). *A commonsense approach to story understanding*. PhD thesis, Massachusetts Institute of Technology.
- Witbrock, M. J., Matuszek, C., Brusseau, A., Kahlert, R. C., Fraser, C. B., and Lenat, D. B. (2005). Knowledge begets knowledge: Steps towards assisted knowledge acquisition in cyc. In *AAAI Spring Symposium: Knowledge Collection from Volunteer Contributors*, pages 99–105.
- Wu, J., Xie, R., Liu, Z., and Sun, M. (2016). Knowledge representation via joint learning of sequential text and knowledge graphs. *arXiv preprint arXiv:1609.07075*.
- Wu, W., Li, H., Wang, H., and Zhu, K. Q. (2012). Probase: A probabilistic taxonomy for text understanding. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 481–492. ACM.
- Xiao, J., Hays, J., Ehinger, K. A., Oliva, A., and Torralba, A. (2010). Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 3485–3492. IEEE.
- Xie, R., Liu, Z., Jia, J., Luan, H., and Sun, M. (2016). Representation learning of knowledge graphs with entity descriptions. In *AAAI*, pages 2659–2665.
- Yamada, I., Shindo, H., Takeda, H., and Takefuji, Y. (2016). Joint learning of the embedding of words and entities for named entity disambiguation. *arXiv preprint arXiv:1601.01343*.
- Zang, L.-J., Cao, C., Cao, Y.-N., Wu, Y.-M., and Cun-Gen, C. (2013). A survey of commonsense knowledge acquisition. *Journal of Computer Science and Technology*, 28(4):689–719.

- Zhendong, D. and Qiang, D. (2006). *HowNet And The Computation Of Meaning (With Cd-rom)*. World Scientific.
- Zhila, A., Yih, W.-t., Meek, C., Zweig, G., and Mikolov, T. (2013). Combining heterogeneous models for measuring relational similarity. In *HLT-NAACL*, pages 1000–1009.
- Zhong, H., Zhang, J., Wang, Z., Wan, H., and Chen, Z. (2015). Aligning knowledge and text embeddings by entity descriptions. In *EMNLP*, pages 267–272.