AMAÇ HERDAĞDELEN

COLLECTING COMMON SENSE FROM TEXT AND PEOPLE

COLLECTING COMMON SENSE FROM TEXT AND PEOPLE

AMAÇ HERDAĞDELEN

In partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Cognitive Sciences
University of Trento

November 2010

Amaç Herdağdelen: *Collecting Common Sense from Text and People,*
In partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Cognitive Sciences
University of Trento, © November 2010

ABSTRACT

In order to display human-like intelligence, advanced computational systems should have access to the vast network of generic facts about the world that humans possess and that is known as commonsense knowledge (books have pages, grocery has a price, ...). Developers of AI applications have long been aware of this, and, for decades, they have invested in the laborious and expensive manual creation of commonsense knowledge repositories. An automated, high-throughput and low-noise method for commonsense collection still remains as the holy grail of AI.

Two relatively recent developments in computer science and computational linguistics that may provide an answer to the commonsense collection problem are text mining from large amounts of data, something that has become possible with the massive availability of text on the Web, and human computation, which is a workaround technique implemented by outsourcing the "hard" sub-steps of a problem to people. Text mining has been very successful in extracting huge amounts of commonsense knowledge from data, but the extracted knowledge tends to be extremely noisy. Human computation is also a challenging problem because people can provide unreliable data and may lack motivation to solve problems on behalf of researchers and engineers. A clever, and recently popularized, technique to motivate people to contribute to such projects it to pose the problems as entertaining games and let people solve those problems while they play a game. This technique, commonly known as games-with-a-purpose approach, has proved a very powerful way of recruiting laypeople on the Web.

The focus of this thesis is to study methods to collect common sense from people via human computation and from text via text mining, and explore the opportunities in bringing these two types of methods together. The first contribution of my study is the introduction of a novel text miner trained on a set of known commonsense facts. The text miner is called BagPack and it is based on a vector-space representation of concept pairs, that also captures the relation between the pairs. BagPack harvests a large number of facts from Web-based corpora and these facts constitute a – possibly noisy – set of candidate facts.

The second contribution of the thesis is Concept Game, a game with a purpose which is a simple slot-machine game that presents the candidate facts – that are mined by BagPack – to the players. Players are asked to recognize the meaningful facts and discard

the meaningless facts in order to score points. Thus, as a result, laypeople verify the candidate set and we obtain a refined, high-quality dataset of commonsense facts.

The evaluation of both systems suggests that text mining and human computation can work very efficiently in tandem. BagPack acts as an almost-endless source of candidate facts which are likely to be true, and Concept Game taps laypeople to verify these candidates. Using Web-based text as a source of commonsense knowledge has several advantages with respect to a purely human-computation system which relies on people as the source of information. Most importantly, we can tap domains that people do not talk about when they are directly asked. Also, relying on people just as a source of verification makes it possible to design fast-paced games with a low cognitive burden.

The third issue that I addressed in this thesis is the subjective and stereotypical knowledge which constitutes an important part of our commonsense repository. Regardless of whether one would like to keep such knowledge in an AI system, being able to identify the subjectivity and detect the stereotypical knowledge is an important problem. As a case study, I focused on stereotypical gender expectations about actions. For this purpose, I created a gold standard of actions (e.g., *pay bill*, *become nurse*) rated by human judges on whether they are masculine or feminine actions. After that, I extracted, combined, and evaluated two different types of data to predict the gold standard. The first type of data depends on the metadata provided by social media (in particular, the genders of users in a microblogging site like Twitter) and the second one depends on Web-corpus-based pronoun/name gender heuristics. The metadata about the Twitter users helps us to identify which actions are mentioned more frequently by which gender. The Web-corpus-based score helps us to identify which gender is more frequently reported to be carrying out a given action. The evaluation of both methods suggests that 1) it is possible to predict the human gold standard with considerable success, 2) the two methods capture different aspects of stereotypical knowledge, and 3) they work best when combined together.

PUBLICATIONS

Parts of this thesis (ideas, figures, results, and discussions) have appeared previously in the following publications:

A. Herdağdelen, M. Baroni (2010). Stereotypical gender actions can be extracted from Web text (Submited for review).

A. Herdağdelen, M. Baroni (2010). Bootstrapping a Game with a Purpose for Common Sense Collection (Submitted for review).

A. Herdağdelen (2010). Gender differences in tweets: A corpus-based analysis of Twitter. Turunç Workshop on Complex Systems, 30 August-01 September 2010, Marmaris, Turkey.

A. Herdağdelen, M. Baroni (2010). The Concept Game: Better commonsense knowledge extraction by combining text mining and a game with a purpose. Proceedings of AAAI Fall Symposium on Commonsense Knowledge, Arlington, 2010.

A. Herdağdelen and M. Baroni (2009). BagPack: A general framework to represent semantic relations. Proceedings of the EACL 2009 Geometrical Models for Natural Language Semantics (GEMS) Workshop, East Stroudsburg PA: ACL, 33-40.

ACKNOWLEDGMENTS

The part of my dissertation which I thought would be the easiest one to write turned out to be the hardest one. I have been indebted to several people who supported me while I was studying for my doctoral degree and I want to express my gratitude for them in a sincere way – which is kind of hard with cliché expressions of gratitude commonly used in acknowledgments. Nevertheless, this thesis has to come to an end, and I cannot finish it without thanking the people who have been by my side during the last three years.

First of all, I want to thank my thesis advisor Marco Baroni for being the kind of advisor that I want to be in the future. He has been a great mentor, teacher and colleague, all at the same time. He was always open to new ideas and amazingly reachable. The median time that took for him to respond to my emails was 893 seconds, slightly below 15 minutes, computed over the more than 1500 mails that I naggingly sent him during my doctorate.

I am indebted to all members of the Language, Interaction and Computation Laboratory (CLIC) in University of Trento for providing a lively research environment. I also want to thank the administrative staff of the Center for Mind and Brain Sciences (CIMeC), for putting up with my – almost always – last-minute paper-work requests, and the technical staff of CIMeC for kindly responding to my requests promptly and keeping the computer cluster up and running all the time.

I am very grateful to my colleagues and friends Deniz Cem Önduygu and Eser Aygün, for all the stimulating discussions we had, technical help they offered for my work, constantly keeping me exposed to a diverse array of ideas, and being the reason for me to visit Istanbul every time I could. In particular, Eser was the technical oracle that I consulted in times of technical desperation, and Cem single-handedly took care of all my visual design problems related to Concept Game.

I am also indebted to Stella Tsigka and Sharon Shan-Shan Chan for they rendered Rovereto a much more livable town than it is, thanks to the generous friendship they offered. I will miss the poker/movie nights and giving them a ride home.

Most importantly, I would not have been in this position – writing acknowledgments over a completed thesis – if it was not for the unconditional love and support of my family. I want to thank my parents, my little sister Duygu, and my extended family for the haven they provided in İzmir – something I generously bene-

fited from. Finally, I want to express my deepest gratitude for my wife Dilek who was always there for me, in good times and bad times, but mostly my crazily-working, non-responding, grumpy times. She shared every minute and great disappointment and joy I had in my life. I dedicate this thesis to Dilek.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

All great deeds and all great thoughts have a ridiculous beginning.

— Albert Camus, The Myth of Sisyphus

Everyday knowledge, otherwise known as *commonsense knowledge* or shortly as *common sense*, is a vast network of generic facts which nearly every person knows but almost never states explicitly – because of the very assumption that it is already shared by everyone. Such knowledge looks naïve and superficial at first look (bedrooms have floors, grocery has a price, ...), but essentially it is what sets apart human beings and the state-of-the-art artificial intelligence (AI) systems [Lieberman, 2008; Minsky, 2000]. The importance of representing everyday knowledge in a computational system in order to attain human-level intelligence has been acknowledged ever since the early days of AI [McCarthy, 1959] and the area continues to be a hot topic for recent research.

The attacks at the commonsense knowledge problem have ranged from manual creation of commonsense knowledge repositories, such as the Cyc database, to knowledge-poor induction of common sense from the text available on the Web [Banko et al., 2007; Lenat, 1995]. However, the manual method is laborious and expensive while the text mining methods are prone to noise. Banko et al. estimate that about 20% of the millions of generic facts they extracted from the Web with a state-of-the-art large scale information extraction system are wrong [Banko et al., 2007]. From one point view, the difficulties that the text mining methods face are not surprising because, by definition, common sense is not explicitly stated in text, and implicit references to it are hard to detect [Havasi et al., 2007].

Another approach, which relies on human computation, to collecting commonsense knowledge is to recruit laypeople from the Web and have them contribute to a knowledge base. The Open Mind Common Sense project [Speer, 2007] relies on the good will of Web surfing volunteers and has been quite successful at collecting tens of thousands of generic facts from ordinary people. An alternative to volunteer work is that of *games with a purpose* [Von Ahn, 2006], inducing Web surfers to contribute various kinds of useful knowledge while they play and have fun. Recently, social networking sites like Facebook[1] have been

1 http://www.facebook.com

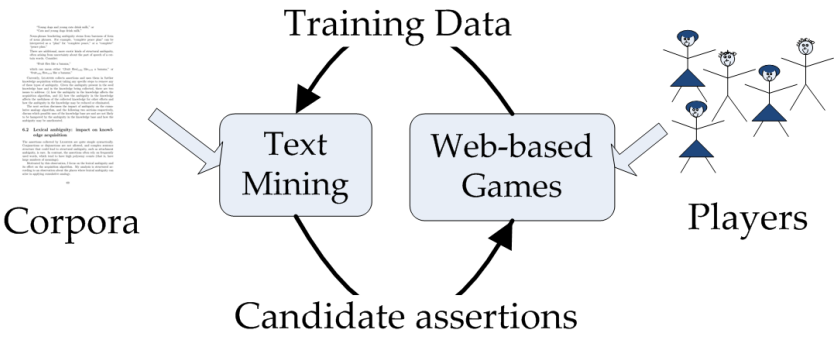

Figure 1: The feedback loop between text mining and human computation. Training data consist of labeled instances of commonsense assertions. The output of text mining is a set of candidate assertions that is further refined in the human-computation stage and then, fed back into the text mining module as training data.

used to deploy such games for easier access to a large user base [Rafelsberger and Scharl, 2009].

Recent advances in the Web technologies made it possible to access large amounts of textual data, while at the same time tap the contributions of numerous Web users. The unifying theme of this thesis is bringing both sources together and making use of the data and the human-computation cycles available on the Web to collect common sense. In the study, I present various methods for common sense collection from text (via text mining) and people (via a game with a purpose), and discuss the advantages of a combined system that brings text mining and human computation together. In addition, I present a novel corpus-based approach that allows us to extend the domain of commonsense knowledge – which is accessible by machines – to social and subjective areas, including stereotypical expectations of people under certain circumstances – in particular gender expectations for daily actions.

Text mining and human computation working in tandem allow us to get the best of both worlds.

The first contribution of this thesis is the introduction of a novel text mining algorithm and a human-computation framework that enable us to attain high throughput (in terms of annotated assertions) while keeping the noise at acceptable levels in commonsense knowledge harvesting. A schematic representation of the integrated system is in Figure 1. The first part of the proposed architecture is BagPack (**Bag**-of-words representation of **Pa**ired **c**oncept **k**nowledge), a vector-space model for representing commonsense assertions (or, more generally, statements about the semantic relation linking two concepts). For a given commonsense relation (e.g. LocationOf, MotivatedByGoal, ...), it evaluates a set of assertions and outputs a list of candidates ranked according to their likelihood of being true. The output of

BagPack is the input of the second part of my proposed architecture, Concept Game, which is a Facebook slot-machine-like game that lets players validate these candidates. The players are shown many "random" assertions and are asked to identify those that make sense in order to gain points.

The abundance of candidate assertions mined by BagPack allows us to free the players from the burden of producing the commonsense facts. All they have to do is to express their assent (or lack of it). I leverage this opportunity by implementing a fast-paced game which does not place a high cognitive load on the player. The candidate assertions extracted from corpora also allow us to employ a more data-driven approach to extend the current knowledge bases; unlike in other human-computation approaches, we can tap the corpora to collect assertions that volunteers/players do not typically provide. In addition, the slot-machine game provides a convenient excuse for the noise in the displayed candidates. Concept Game is a game of chance and a player is expected to see many meaningless assertions before "hitting the jackpot". Thus, the low precision of the text miner module becomes a natural part of the game experience, not a source of frustration. Another advantage is that, besides their inherent value, such cleaned-up data can be used to assess the quality of text mining and fed back to the algorithm as labeled training materials.

The second contribution is the introduction of the notion of *stereotypical common sense* in computational commonsense mining. Stereotypical expectations constitute an important part of our commonsense knowledge. Whether they are right or wrong, artificial intelligence systems should explicitly know that people, in general, *expect* that *men like football* or *women like shopping* [Sherron, 2000]. I propose a corpus-based algorithm which incorporates a large set of personal status messages collected from Twitter[2] and the metadata about the Twitter users to extract the stereotypical gender expectations of commonsense concepts and actions. This is intended to be an example of how the metadata about the users and the user-generated content can be brought together to extend the commonsense knowledge such that it will reflect the more subjective dimensions of common sense.

Commonsense repositories must know about the stereotypical expectations and prejudices of people.

Therefore, I set the following four research questions that I pursue in this manuscript.

RESEARCH QUESTION 1 Do corpora, and Web-based corpora in particular, contain commonsense knowledge, and if so can we extract it?

RESEARCH QUESTION 2 Can we combine text mining and human computation in order to achieve a better commonsense collection?

2 http://www.twitter.com

A corollary and a special case of this question is:

Research Question 3 Is it possible to collect high-quality commonsense assertions from people while they play a fast-paced game where the players' main motivation is having fun?

Research Question 4 Can we extract subjective aspects of common sense like the stereotypical expectations or prejudices of people from Web corpora?

At the end of the discourse, my answer to all four research questions will be "Yes", and this dissertation is intended to present the evidence that supports this conclusion. The organization of the manuscript is as follows.

In Chapter 2, I provide some background on the techniques and materials that are commonly used in subsequent chapters. In particular, I introduce different approaches for representing common sense, and different corpora that I use throughout this study. In Chapter 3, I address the first research question and provide the description of the text mining algorithm that I propose along with its evaluation on two standard semantic tasks where we can compare its performance with the state of the art, and on a third task pertaining to commonsense knowledge extraction. In Chapter 4, I present and discuss the game with a purpose, designed to collect common sense from people. Also in this chapter, the performance of bootstrapping the text miner with the output of the game is evaluated. The methods that I use to extract stereotypical gender expectations from corpora are detailed in Chapter 5. Finally, Chapter 6 concludes the manuscript with the main achievements of the current study and future directions.

2 BACKGROUND

Jerry: *Oh, one more thing about the car.*
Let it warm up for a minute.
George: *That's a tough minute. It's like*
waiting in the shower for the conditioner to work.

— Seinfeld, The Busboy

2.1 REPRESENTATION OF COMMON SENSE

Common sense consists of the assumptions and known facts about daily life shared by a great majority of people. The ontologist Barry Smith stresses the enormous amount of common sense: "Common sense includes a massive storehouse of factual knowledge about colours and sounds, about time and space, about what foods are edible and what animals are dangerous." [Smith, 1995]. For an average citizen of the modern world, we can easily extend this definition to cover facts like "One should turn off his cellphone while attending a concert." or "People eat breakfast in the morning." [Lieberman, 2008]. It looks tricky to appeal to majority for a definition ("What exactly is a great majority of people?"), but this appeal is the very essence of the functional benefits of common sense. It allows us to communicate with other people without being verbose to the point that communication is impossible. For an efficient communication, we can rely on the set of beliefs and facts we share with our interlocutor [Lenat, 1996].

The acquisition and representation of commonsense knowledge are at the core of AI research and there have been several studies that address these problems. One of the earliest and best-known research projects that attempt to solve the commonsense problem is the Cyc project initiated by Douglas Lenat in 1984 [Lenat, 1995]. Today, Cyc is a commercial knowledge base (owned and developed by Cyccorp Inc.[1]) that intends to capture a significant portion of commonsense knowledge. It employs a detailed ontology of concepts, actions and rules, and has its own special formal language to enter new facts. The underlying semantics extends beyond a first-order predicate logic to allow an expressive representation. ResearchCyc is a limited version of Cyc available under a free license and – as of 2005 – it contained

1 http://www.cyccorp.com/

approximately a million assertions about more than a hundred thousand symbols [Ramachandran et al., 2005]. The fact that Cyc is a formal repository of commonsense knowledge increases the amount of effort to construct, extend, and maintain it because one needs to know about the formal semantics and syntax underlying the knowledge base.

The Open Mind Common Sense[2] (OMCS) project is another commonsense database project initiated by MIT's Media Lab in 1999 [Speer, 2007]. A key difference between Cyc and OMCS is that the latter relies on semi-structured snippets of natural language phrases to represent commonsense knowledge instead of a logical formalism. For example, an instantiation of the commonsense relation of *AtLocation* between the two concepts *ashtray* and *bar* can be represented as "Something you find at a *bar* is an *ashtray*". Here, the template "Something you find at a {} is an {}." corresponds to *AtLocation* and the concepts are normalized forms of natural language phrases (e.g., "ashtrays", "an ashtray", and similar phrases are normalized to *ashtray* by means of lemmatization and stop-word removal.

Representing commonsense knowledge as semi-structured natural language phrases makes it possible to tap laypeople's efforts to build a commonsense database.

OMCS's choice of semi-structured phrases reduces the deductive inference power of the system because of the ambiguities introduced by natural language and the imperfect mapping from phrases to concepts. On the other hand, the less formal representation allows to recruit laypeople to enter facts into OMCS because almost no training is required to formulate facts. In fact, the entire content of OMCS is based on the volunteers' efforts. Since 1999, more than 16,000 people contributed to OMCS via a Web-based interface, resulting in more than 700,000 English facts [Havasi et al., 2009]. Moreover, the ambiguity and redundancy introduced by the natural language representation can be an advantage for flexible inference. Interested readers can refer to Liu and Singh [2004] for a detailed discussion of the advantages and disadvantages of the natural language representation.

ConceptNet is a semantic network that is based on the facts contained in OMCS [Liu and Singh, 2004]. The vertices are the concepts and the links come from a closed set of commonsense relations such as *AtLocation*, *CapableOf*, and *HasLastSubEvent*. ConceptNet serves as the seed of our text miner as explained in Section 3.5.1.

2.2 CORPORA

In computational linguistics, collections of unstructured texts – called corpora – are used for statistical analyses, hypothesis testing, or exploratory purposes. A corpus is usually sampled

2 http://openmind.media.mit.edu/

from a specific source (e.g. the works of a writer, newspapers, etc.) and it is meant to be a representative sample of that source. Since our interest in this study is in employing real-world text samples as they occur on the Web, I used the following three corpora.

- *ukWaC*, a corpus consisting of two billion tokens and obtained from a linguistically-informed crawl of the *uk* domain conducted between 2005 and 2007, automatically annotated with part-of-speech and lemma information using the TreeTagger tool [Baroni et al., 2009].
- *Wikipedia*[3], 2009 dump of the English Wikipedia which contains around 800 million tokens, POS-tagged and lemmatized with the same tools used for ukWaC.
- The Edinburgh Twitter Corpus (*ETC*), a corpus of 97 million tweets and approximately two billion tokens, randomly sampled from the Twitter public timeline during a 4-month period spanning November 11th 2009 until February 1st 2010 [Petrović et al., 2010].

Both ukWaC and Wikipedia are already linguistically pre-processed, and I used the lemmatized versions of the tokens, discarding the POS tags. On the other hand, ETC consists of very colloquial text (e.g. *@mr_smile lol y'all be safe, ppl drivin hella slow this morning!! #traffic Grrr fat foot no likey :)*) and the linguistic tools that are used for ukWaC and Wikipedia are not particularly suitable for this kind of text. Instead, for ETC, I opted for the following ad hoc post-processing, adapted for Twitter content. First, the content is lowercased and tokenized by the StandardAnalyzer class of the Lucene search engine library[4]. During this step, no stop-word list is employed; some common emoticons (e.g., ":)", ":<", ...), the references to other Twitter users (user names appended by "@"; e.g., "@mr_smile"), and hash tags that are used to tag the tweets (keywords appended by "#"; e.g. "#traffic") are kept intact. After the tokenization, the tokens are "lemmatized" by using a token-lemma look-up table constructed from the pre-processed ukWaC: For each token, its most lemmatized form is found in ukWaC and that lemma is used. However, only the token-lemma associations that are observed at least 100 times in ukWaC are allowed in the look-up table. Any token that is not found in the table is tagged as *unobserved* and kept as is without any lemmatization.

The content in Twitter is very colloquial and may have a peculiar writing style; therefore, it requires special linguistic post-processing.

In the construction of ETC, no attempts were made to distinguish or filter according to the language of tweets. Therefore,

3 http://wacky.sslmit.unibo.it
4 http://lucene.apache.org/java/docs/index.html

it is linguistically spurious. Since I am interested mainly in the English language, a simple language filter is employed to make sure that only English tweets are processed: The filter discards a tweet if it does not contain at least four tokens that are observed in the ukWaC-based token-lemma table (to make sure it is long enough) or which contains more than 20% unobserved tokens (to make sure it does not contain many non-English words). This step results in a set of 34 million tweets that are likely to be in English.

A summary of the key properties of the corpora is given in Table 1.

	UKWAC	WIKIPEDIA	ETC	ETC-ENGLISH
Source	Web	Wikipedia	Twitter	Twitter
Language	English	English	Multi	English
Size (# tokens)	2 billion	800 million	2 billion	500 million

Table 1: A summary of the corpora used in this thesis.

3

BAGPACK

George: *Plus, they give you those word association tests. I love those.*
Jerry: *That'd be great. There's no wrong answer.*

— Seinfeld, The Truth

In this chapter, I propose an automated text mining method, BagPack, to extract commonsense knowledge from corpora. The proposed method is first evaluated on two classic semantic tasks: solving analogy questions and predicting selectional preference in verbs. I employ these two tasks in particular because, besides being popular tasks in computational semantics [Baroni and Lenci, 2010; Biçici and Yuret, 2006; Padó et al., 2007; Turney, 2006a; Turney and Littman, 2005], recognizing analogies (e.g., the sole of a foot is similar to the palm of a hand) and predicting the object/verb preferences of verbs (e.g., knowing that one can shoot a deer, but a deer is unlikely to shoot) are among the tasks that require commonsense knowledge as well. Finally, I evaluate the model also on the task of extracting commonsense assertions of the sort attested in ConceptNet.

In the subsequent sections, I first review the previous work on automated techniques for commonsense knowledge extraction from text and then describe BagPack. Then, I discuss the performance of BagPack on the two distributional semantic tasks and on the third task of extracting commonsense.

3.1 RELATED WORK

Corpus-based techniques proved themselves reliable methods of knowledge extraction by using statistical analysis of frequently occurring patterns in large amounts of text. They are widely adapted in domains ranging from machine translation to biomedical text mining [Buitelaar and Cimiano, 2008].

Can we follow a corpus-based path for the commonsense problem and mine common sense from Web corpora? At first look, the answer should be no because, by definition, commonsense facts go unstated explicitly in discourses. A robber does not start his threat by reminding his victim that "guns shoot bullets" and "bullets kill people". Rather, he simply says "Give me your wallet or I will shoot!" (assuming, for the sake of argument, that we have a corpus of robberies). As a matter of fact, precisely this observation motivates the effort for constructing commonsense

knowledge bases – if common sense was readily found in text (context in general) no one would have to store it explicitly!

Nonetheless, whether it is possible to extract general – and of course true – facts about the daily life from corpora is an empirical question, and the literature provides encouraging results on this problem. A key observation is that many types of commonsense knowledge are reflected at the surface level in text. Lucy Vanderwende (2005) cites the following example [Vanderwende, 2005]:

Although common sense is not explicitly stated in text, it is still possible to infer useful knowledge by using some linguistic cues.

> A bat is the only mammal that can truly fly.

Upon reading this sentence, even if one knows nothing about what a mammal or bat is, one can deduce that

- bats can fly
- mammals (mostly) do not fly.

We can easily add to the list that *some mammals are not bats*, *some mammals (other than bats) can look like they fly*, etc. Therefore, algorithms that leverage such implicit clues can be employed to extract commonsense knowledge. Vanderwende herself presents a proof of concept to show how a system that relies on lexico-syntactic heuristics can be used to expand a given commonsense database.

In another related study, Strohmaier and Kröll (2009) analyze the search query logs released by two commercial search engines and conclude that "search query logs are a potential source of common human goals." [Strohmaier and Kröll, 2009]. Their methodology is to compare the verb phrases extracted from query logs to the verb phrases already contained in ConceptNet (e.g., "gain weight", "make paper airplane") A significant amount of overlap between the goals extracted from corpus and the goals contained in ConceptNet motivates their conclusion.

The two aforementioned studies tell us how and why corpus-based approaches can succeed in the task of commonsense knowledge extraction. Lucy Vanderwende's study shows that there is a certain amount of commonsense knowledge hidden beneath the surface level linguistic expressions: people express such knowledge implicitly while they talk about other things. Strohmaier and Kröll present a novel way to collect knowledge about the actions that are pertinent to people's daily life: in certain settings – like searching for information on the Web – people may provide explicit cues about what is important to them. The following studies are examples of the systems that actually attempt to carry out the task of extracting common sense.

KNEXT (*Kn*owledge *Ex*traction from *T*ext) is a system proposed for extracting "general world knowledge from miscellaneous texts, including fiction" [Schubert and Tong, 2003]. Schubert and Thong (2003) provide an extensive evaluation of the output of KNEXT on the British National Corpus (a balanced and representative collection of spoken and written English samples, containing 100 million words)[1] and according to their evaluation based on five human judges, almost 60% of the generated propositions are found to be "reasonable general claims" by any given judge. However, the agreement between judges on individual facts is not extremely high. Considering only cases where all five judges agree, the ratio of "reasonable general claims" reduces to one third. Recently, it was also shown by Schubert and collaborators that – with significant post-processing and filtering of output – more noisy corpora such as text coming from blogs can be used as another source of commonsense knowledge [Gordon et al., 2010b].

Using Web-based corpora as a source for commonsense knowledge is indeed becoming a popular technique. The abundance of text coming from blogs, Web pages, discussions in newsgroups and other social media (e.g. Twitter, Facebook) allows us to cover a wide array of domains and extract a great number of facts. The sheer amount of input also helps to fight with the noise in the output of knowledge extraction systems. It is possible to apply very strict filtering conditions and focus on a – relatively – small subset of the output to obtain high quality facts with a trade-off in recall (the amount of extracted knowledge relative to the amount that is actually contained in the corpus). In the following paragraphs, I discuss some examples of this approach.

In 2005, a research group from Cyccorp reported a preliminary experiment which extends the knowledge base of Cyc by issuing template-based queries to a commercial search engine and subsequently analyzing the results [Matuszek et al., 2005]. The first step in their approach is to identify missing pieces of knowledge in Cyc such as the missing information about the founder of the Palestine Islamic Jihad Organization, represented as a tuple *(foundingAgent, PalestineIslamicJihad, ?WHO)*. In subsequent steps, they reformulate the missing information as a natural language template such as "PIJ, founded by *", issue a search to Google[2] by using this template as the query, and analyze the resulting snippets. An example snippet returned by the search engine is "PIJ founder Bashir Musa Mohammed Nafi is still at large..." which suggests that *Bashir Musa Mohammed Nafi* is a likely candidate to be inserted into the initial incomplete tuple. After some post-

1 http://www.natcorp.ox.ac.uk/
2 http://www.google.com

processing and filtering – which includes consistency checking with the already known facts in Cyc – the final candidates are presented to a human volunteer or an expert, and those that pass the final evaluation are scheduled to be inserted in Cyc. In their evaluation, approximately 94% of the mined facts were discarded in the post-processing and filtering stage (e.g., they were already found in Cyc, or they were not compatible with the existing facts in Cyc, etc.). Of the 6% that passed the filtering, approximately half were found to be true by a human reviewer.

The ConceptMiner of Ian Eslick is another system that populates commonsense assertions by issuing pattern-based queries to Google [Eslick, 2006]. The system is based on ConceptNet and the aim is to extend ConceptNet by finding new pairs of concepts that are related to each other by one of the commonsense relations contained in ConceptNet. The key idea in ConceptMiner is similar to the one discussed in Vanderwende (2005); a set of known facts contained in ConceptNet serves as a seed and ConceptMiner extracts typical surface-level linguistic expressions that contain pairs of concepts bound by a given relation. For instance, for the pair of concepts *dog* and *bark* which are related to each other by *CapableOf*, a search operation can result in the following phrases: "when a dog barks", "the dog never stopped barking", etc...[3] Then, another search session is used to find further instantiations of those surface forms to get new candidates of concept pairs for the relation (e.g. the queries "when a {} {}" or "the {} never stopped {}" are issued and the results are parsed to extract new pairs of concepts). Statistical analysis and extensive filtering of the candidates result in a substantially smaller but high-quality set of assertions that can be inserted in ConceptNet.

Another application which uses ConceptNet as a seed and attempts to extend it by using the Web as a corpus is that of Yu and Chen, presented in 2010 [Yu and Chen, 2010]. In this approach, for each relation in question, a dataset of assertions is constructed from the facts stored in ConceptNet. In addition to the original ConceptNet assertions (which serve as positive training instances), an equal number of randomly-crafted assertions are added to the datasets to serve as negative instances. Then, each instance (i.e., each pair of concepts) in a training set is represented in a vector space that captures the co-occurrence patterns in a corpus (Yu and Chen use the Google Web 1T 5-Gram corpus [Brants and Franz, 2006]). The results show that a support vector machine (SVM) trained on a subset of the initial dataset is able to discriminate meaningful facts from random facts in the left-out part of the dataset with an accuracy that is significantly

3 This is a very simplified example. In the real setting, ConceptMiner employs other analyses like POS-tagging and lemmatization.

higher than random performance. The accuracy of the trained SVMs range from 55% to 85% for different relations and model parameters (where 50% is the random baseline).

The work of Yu and Chen is one example of more general corpus-based distributional methods. There has been much previous work on corpus-based models to extract broad classes of related words. The literature on word space models [Sahlgren, 2006] has focused on taxonomic similarity (synonyms, antonyms, co-hyponyms,...) and general association (e.g., finding topically related words), exploiting the idea that taxonomically similar or associated words will tend to occur in similar contexts, and thus share a vector of co-occurring words. The literature on relational similarity, on the other hand, has focused on *pairs* of words, devising various methods to compare how similar the contexts in which target pairs appear are to the contexts of other pairs that instantiate a relation of interest [Pantel and Pennacchiotti, 2006; Turney, 2006a, 2008]. As an example, Turney's Latent Relational Analysis (LRA) achieves human-level performance in the SAT analogy questions[Turney, 2006a].

3.2 THE BAGPACK MODEL

The model I propose is called BagPack (*Bag*-of-words representation of *Pa*ired *c*oncept *k*nowledge). BagPack's approach to the extraction of semantic information is to construct a vector-based representation of a pair of terms in such a way that the vector represents both the contexts where the two terms co-occur (that should be very informative about their relation) and the contexts where the single terms occur on their own (possibly less directly informative but also less sparse).

Depending on the task at hand, a context may be defined as a fixed-length window around a concept, the sentence or even as the document that contains the concept.

An illustration of how the vectors are constructed is given in Figure 2. For a given pair of concepts, BagPack constructs three different sub-vectors, one for the first term (recording frequency of co-occurrence of the first term with context items which may be unigrams or n-grams depending on implementation), one for the second term (with the same kind of information), and one for the co-occurring pair (keeping track of the items that occur in sentences where both terms occur). The concatenation of these three sub-vectors is the final vector that represents the pair.

Before going into further details, we need to know what a "co-occurrence" precisely means, define the notion of context, and determine how to structure the vector.

Let (W_1, W_2) denote an ordered pair of words W_1 and W_2. We say the two words *occur as a pair* whenever one of the following pseudo regular expressions is observed in the corpus: "$L\ W_1\ B\ W_2\ R$" or "$L\ W_2\ B\ W_1\ R$" where L and R can be empty

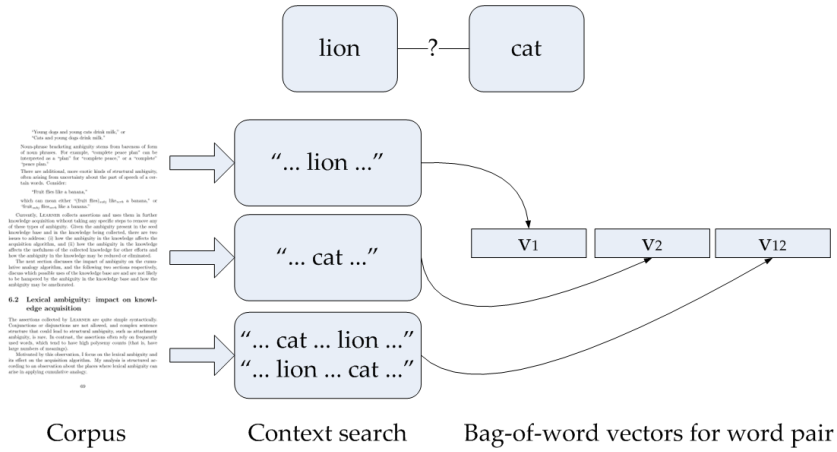

Figure 2: An example vector construction for the pair *(lion, cat)*. The resulting vector is a concatenation of three sub-vectors each of which represents a different aspect of the pair. The details of the context searches are given in the text.

strings or concatenations of up to 2 words and similarly, B can be either an empty string or concatenation of up to 5 words (i.e. $L_1, \ldots, L_i$, $R_1, \ldots, R_j$, and $B_1, \ldots, B_k$ where $i, j \leq 2$ and $k \leq 5$). Together, these two patterns constitute the *pair context* for W_1 and W_2. The patterns are matched with the longest possible substring while making sure that B does not contain W_1 nor W_2. The basis terms that are observed in all contexts which match with these patterns are used to construct the *paired-occurrence sub-vector*, denoted by $\mathbf{v_{1,2}}$.

BagPack is a vector space model that represents pairs of concepts as vectors.

The *single occurrence sub-vector* is constructed in a similar way. For a single word W, the following pseudo regular expression identifies an observation of *occurrence*: "$L\ W\ R$" where, in this case, L and R can be empty strings or concatenations of up to 4 words separated by whitespace (i.e. $L_1, \ldots, L_i$ and $R_1, \ldots, R_j$ where $i, j \leq 4$). Each observation of this pattern constitutes a *single context* of W. The pattern is matched with the longest possible substring without crossing sentence boundaries and the basis terms in the matching contexts are used to construct the occurrence sub-vector. For the given pair (W_1, W_2), two single occurrence sub-vectors $\mathbf{v_1}$ and $\mathbf{v_2}$ are created by substituting W in the pattern with W_1 and W_2 respectively.

In the end, the final vector that represents the ordered pair (W_1, W_2) is the concatenation of the single-occurrence and paired-occurrence sub-vectors, $\mathbf{v_1 v_2 v_{1,2}}$. The number of context words allowed before, after, and between the targets are actually model parameters but for the experiments reported in this study, I used the aforementioned values with no attempt at tuning.

The population of BagPack starts by identifying the b most frequent unigrams in the corpus as *basis terms*. Let T denote a basis

term. For the construction of $\mathbf{v_1}$, I define two features for each term T: t_{pre} corresponds to the number of observations of T in the single contexts of W_1 occurring before W_1 and t_{post} corresponds to the number of observations of T in the single occurrence of W_1 where T occurs after W_1 (i.e. number of observations of the single contexts where $T \in L$ and $T \in R$ correspondingly). The construction of $\mathbf{v_2}$ is identical except that this time the features correspond to the number of times the basis term is observed before and after the target word W_2 in single contexts. The construction of the paired-occurrence sub-vector $\mathbf{v_{1,2}}$ proceeds in a similar fashion but, in addition, also the order of W_1 and W_2 as they co-occur in the pair context is incorporated into the representation: The number of observations of the pair contexts where W_1 occurs before W_2 and T precedes (follows) the pair, are represented by feature t_{+pre} (t_{+post}). The number of cases where the basis term is in between the target words is represented by t_{+betw}. The number of cases where W_2 occurs before W_1 and T precedes the pair is represented by the feature t_{-pre}. Similarly the number of cases where T follows (is in between) the pair is represented by the feature t_{-post} (t_{-betw}).

Assume that the words "only" and "that" are our basis terms and consider the following context for the word pair ("cat","lion"): "Lion is the only cat that lives in large social groups." The observation of the basis terms should contribute to the paired-occurrence sub-vector $\mathbf{v_{1,2}}$ and since the target words occur in reverse order, this context results in the incrementation of the features $only_{-betw}$ and $that_{-post}$ by one.

To sum up, we have b basis terms. Each of the single-occurrence sub-vectors $\mathbf{v_1}$ and $\mathbf{v_2}$ consists of $2b$ features: Each basis term gives rise to 2 features incorporating the relative position of basis term with respect to the single word. The paired-occurrence sub-vector, $\mathbf{v_{1,2}}$, consists of $6b$ features: Each basis term gives rise to 6 new features; $\times 3$ for possible relative positions of the basis term with respect to the pair and $\times 2$ for the ordering of the pair.

BagPack is an umbrella name that is used for a set of algorithms that rely on the same vector space.

It is worth to note that in this manuscript, BagPack serves as an umbrella name for a family of algorithms all of which depend on the semantic vector space that we just discussed. As we will see in the subsequent sections, BagPack may be employed as part of a supervised algorithm where the vectors represent labeled training instances and a machine learning algorithm is used to classify or rank unknown instances in a test set, or it may be used in an unsupervised model where the distances between the vectors in this space is used to assess semantic similarity.

The specifics of the adaptation to each task will be detailed when I describe the experiments.

3.3 SAT ANALOGY RECOGNITION TASK

The SAT analogy questions task was introduced by Turney et al. [2003]. In this task, there are 374 multiple choice questions with a pair of related words as the *stem* (e.g., *ostrich-bird*) and 5 other pairs as the *choices* (e.g., *lion-cat*, *goose-flock*, *ewe-sheep*, *cub-bear*, *primate-monkey*). The correct answer is the choice pair which has the relationship most similar to that in the stem pair (*lion-cat* in this example).

3.3.1 *Experimental setup*

SOURCE CORPORA For this task, BagPack vectors are constructed by using the Web-derived English Wikipedia and ukWaC corpora, about 2.8 billion tokens in total (for details refer to Chapter 2.2).

MODEL IMPLEMENTATION During the implementation, I did not carry out a search for "good" parameter values. Instead, the model parameters are generally picked at convenience to ease memory requirements and computational efficiency. The basis terms are the most frequent 5000 lemmas plus the words that are occurring in the SAT analogy task (i.e., $b = 6072$). Once the BagPack vectors are computed for all pairs in the dataset, we get a *co-occurrence matrix* with pairs on the rows and the features on the columns (including pair- and single-occurrence features). Pointwise-mutual-information (*PMI*) feature weighting is applied to the co-occurrence matrix [Church and Hanks, 1990]. PMI is the log-ratio of the observed joint probability of two random events to their expected joint probability under the independence assumption.

PMI, also known as the specific mutual information, helps to distinguish the word pairs with high co-occurrence counts just because the individual words are very common from the ones where there is really a connection between the pair of words.

Because the task consists of individual multiple-choice questions to be answered independently, employing an unsupervised method is a natural choice and I adopt an unsupervised approach to answer the SAT questions. The cosine similarities between the choices and the stem pair are computed for each question and the choice that is the most similar to the stem is picked as the predicted answer.

3.3.2 *Results*

I evaluated the model for the following representations: 1) Single-occurrence sub-vectors ($\mathbf{v_1 v_2}$ condition) 2) Paired-occurrence sub-vectors ($\mathbf{v_{1,2}}$ condition) 3) Entire co-occurrence matrix ($\mathbf{v_1 v_2 v_{1,2}}$ condition).

A question is said to be *complete* when all of the related pairs (i.e., stem and choices) are represented by non-zero vectors. The *coverage* is defined as the percentage of complete questions and *accuracy* as the percentage of complete questions that are answered correctly.

The results are given in Table 2. The $\mathbf{v_{1,2}}$ only condition achieves a relatively low coverage: only 250 questions out of 374 are complete of which 39.6% are answered correctly. Apart from the low coverage, among the questions that are answered, the performance of the three conditions are very similar.

CONDITION	ACCURACY	COVERAGE
$\mathbf{v_{1,2}}$	39.6%	66.8%
$\mathbf{v_1 v_2}$	38.3%	99.5%
$\mathbf{v_1 v_2 v_{1,2}}$	39.6%	100.0%

Table 2: Percentage of correctly answered questions in SAT analogy task.

In Table 3, the best accuracy results we observe for BagPack are compared to previous studies.[4] Among these models, *DM and LRA-10 are of particular interest because they are corpus-based methods that are trained on ukWaC and Wikipedia – the same corpora that I used for training BagPack. Note that LRA-10 is a re-implementation of the Latent Relational Analysis method of Turney [2006a] (i.e. LRA-06), by Baroni and Lenci [2010].

Overall, the performance of BagPack is not at the top of state of the art – or close to the human performance of 57% for that matter – but comparable to that of most recent systems developed for the SAT challenge.

3.4 SELECTIONAL PREFERENCE TASK

Linguists have long been interested in the semantic constraints that verbs impose on their arguments, a broad area that has also attracted computational modeling, with increasing interest in purely corpus-based methods [Erk, 2007; Padó et al., 2007]. This task is of particular interest to us as an example of a broader class of linguistic problems that involve *productive* constraints on composition. As has been stressed at least since Chomsky's early work [Chomsky, 1957], no matter how large a corpus is, if a phenomenon is productive there will always be new well-formed instances that are not in the corpus. In the domain of selectional restrictions this is particularly obvious: we would not say that

4 See http://aclweb.org/aclwiki/ for further information and references.

MODEL	ACCURACY	95% CI	MODEL	ACCURACY	95% CI
Human[1]	57.0	52.0-62.3	LSA[7]	42.0	37.2-47.4
LRA-06[2]	56.1	51.0-61.2	*BagPack*	39.6	34.6-44.7
PERT[3]	53.3	48.5-58.9	LRA-10[6]	37.8	32.8-42.8
PairClass[4]	52.1	46.9-57.3	PMI-IR-06[2]	35.0	30.2-40.1
VSM[1]	47.1	42.2-52.5	DepDM[6]	31.4	26.6-36.2
k-means[5]	44.0	39.0-49.3	LexDM[6]	29.3	24.8-34.3
TypeDM[6]	42.4	37.4-47.7	Random	20.0	16.1-24.5

Table 3: Comparison of accuracy in solving SAT analogies with 95% binomial confidence intervals. Model sources: [1][Turney and Littman, 2005]; [2][Turney, 2006a]; [3][Turney, 2006b]; [4][Turney, 2008]; [5][Biçici and Yuret, 2006]; [6][Baroni and Lenci, 2010]; [7][Quesada et al., 2004]

an algorithm learned the constraints on the possible objects/patients of eating simply by producing the list of all the attested objects of this verb in a very large corpus; the interesting issue is whether the algorithm can detect if an unseen object is or is not a plausible "eatee"/"eater", like humans do without problems. I consider this aspect of the selectional restriction relevant to the commonsense problem. After all, it is commonsense knowledge that a mushroom cannot eat a human. However, we cannot rely on the hope that such facts are explicitly stated in a corpus.

Specifically, I evaluate the performance of a BagPack representation on the dataset constructed by Padó [2007] who collected average plausibility judgments (from 20 speakers) for nouns as either subjects or objects of verbs (211 noun-verb pairs).

3.4.1 *Experimental Setup*

I formulate this task as a regression problem. I train a ϵ-SVM regressor [Canu et al., 2005] with 18-fold cross validation: Since the pair instances are not independent but grouped according to the verbs, one fold is constructed for each of the 18 verbs used in the dataset. In each fold, all instances sharing the corresponding verb are left out as the test set. The performance measure for this task is the Spearman correlation between the human judgments and the model's estimates. There are two possible ways to calculate this measure. One is to get the overall correlation between the human judgments and the estimates obtained by concatenating the output of each cross-validation fold. That measure allows us to compare the method with the previously reported results and it is analogous to micro-averaging the results (i.e. "micro-aggregated" Spearman correlation) [Yang and Liu, 1999]. However, it cannot control for a possible verb-effect on the human judgment values: If the average judgment values of the pairs associated with a specific verb is significantly higher (or lower) than the average of the pairs associated with another verb, then any regressor which simply learns to assign the average value to all pairs associated with that verb (regardless of whether there is a patient or agent relation between the pairs) will still get a reasonably high correlation because of the variation of judgment scores across the verbs. To control for this effect, I also calculated the correlation between the human judgments and the estimates for each verb's plausibility values separately, and computed the average Spearman correlations over all verb groups. I report this "macro-aggregated" results as the "mean" results.

3.4.2 *Results*

The coverage for this dataset is quite high. All pairs are represented by non-zero vectors but two, which are discarded in subsequent experiments.

The model is evaluated for the three different conditions on vector representation: 1) Single-occurrence sub-vectors ($\mathbf{v_1v_2}$ condition), 2) Paired-occurrence sub-vectors ($\mathbf{v_{1,2}}$ condition), and 3) Entire co-occurrence matrix ($\mathbf{v_1v_2v_{1,2}}$ condition).

The results are given in Table 4, and the best results we see are an overall correlation of 0.45 and a mean correlation of 0.36, both in the combined case $\mathbf{v_1v_2v_{1,2}}$. In the same table, we also see several previously reported results on the same dataset, suggesting that a supervised algorithm based on the the BagPack representation is capable of reaching a state-of-the-art level for this task.

METHOD	COVERAGE (%)	OVERALL	MEAN
$\mathbf{v_{1,2}}$	97	0.38	0.31
$\mathbf{v_1v_2}$	98	0.37	0.25
$\mathbf{v_1v_2v_{1,2}}$	98	0.45	0.36
TypeDM[1]	100	51	-
Padó[2]	97	51	-
ParCos[2]	98	48	-
DepDM[1]	100	35	-
LexDM[1]	100	34	-
Resnik[2]	98	24	-

Table 4: Spearman correlations between the targets and estimations for selectional preference task. Model sources: [1]Baroni and Lenci [2010]; [2]Padó et al. [2007].

3.5 MINING FOR COMMON SENSE

We have seen that BagPack obtains reasonable performance on two semantic tasks that are also pertinent to common sense. The composite nature of the BagPack vectors allow us to represent pairs of concepts as co-occurrence vectors even if the pairs are not co-occurring the corpus.

The paired-occurrence vector presumably captures the relation between a pair of vectors more accurately because it directly depends on the contexts in which the pair co-occurs. However, for rarely stated concepts this may lead to a very sparse represen-

tation for which many pairs have zero vectors. In this case, the single-occurrence sub-vectors offer us a fall-back representation in which the individual occurrence contexts of the concepts are represented.

Even though two concepts do not co-occur in a corpus, BagPack can provide a reasonable fall-back representation of them, thanks to the single-occurrence sub-vectors.

In both tasks, the combined vector representation – which concatenates the paired-occurrence and the two single-occurrence sub-vectors together – obtained the widest coverage and best performance measures. Motivated by these observations, I picked BagPack with the combined vector representation as the model of my choice, to be used to attack the commonsense extraction problem.

In this section, we turn our attention to commonsense mining, with the first of a series of experiments in which I attempt to extract commonsense assertions of the sort that are stored in ConceptNet.

The training examples fed to BagPack come from ConceptNet, whereas evaluation is carried on a candidate set of assertions mined from Wikipedia. In this task, I use a c-SVM [Canu et al., 2005] that is trained on labeled examples of the *training set* (i.e., ConceptNet-based assertions) and the confidence scores of the SVM on the *test set* (i.e., Wikipedia-based candidate assertions) are used to rank and evaluate the performance.

3.5.1 *Training materials*

The initial training datasets are based on the assertions contained in ConceptNet 4[5]. In this study, I focus on five relations that represent rather different ways in which concepts are connected and correspond to more (IsA) or less (SymbolOf) traditional ontological relations, and tend to link words/phrases from different syntactic classes: IsA (*cake, dessert*); AtLocation (*cake, oven*); HasProperty (*dessert, sweet*); MotivatedByGoal (*bake cake, eat*); SymbolOf (*Sacher Torte, Vienna*).

Half of each training dataset consists of original ConceptNet assertions while the other half consists of randomly-crafted versions of original assertions to make sure the dataset contains negative instances.

The training datasets of each relation consist of approximately 500 assertions. Half of the assertions (SymbolOf is instantiated by 151 assertions only, and I use them all) were randomly sampled from ConceptNet and the remaining assertions are constructed as bogus assertions by randomly picking an original assertion from the first half (e.g., *Sacher Torte SymbolOf Vienna*) and changing i) either one of its associated concepts with a random concept from ConceptNet (e.g., *Sacher Torte SymbolOf win election*), or ii) the original relation with another of the five relations I work with (e.g., *Sacher Torte IsA Vienna*).

For annotation of the training dataset, a total of 22 expert raters were recruited, all advanced students or researchers in

5 http://conceptnet.media.mit.edu/

artificial intelligence, semantics or related fields. The raters were given precise instructions on the purpose of the procedure and had to annotate assertions as *meaningful* or *meaningless*. For each rater, the probability of agreement with the majority vote on a random assertion was computed; and as a precaution to ensure high-quality data, I discarded the responses of five raters with a probability lower than 0.70. Only the 2,051 assertions which received at least two meaningful or two meaningless responses were considered for further analysis. The final label of an assertion was decided by the majority vote and the ties were broken in favor of meaninglessness. Table 5 summarizes the annotation results for each relation. Note that some of the original assertions coming from ConceptNet were rated as meaningless (for example: *bread IsA put butter*; *praise IsA good job*; *read newspaper MotivatedByGoal study bridge*). These assertions should serve as high-quality negative instances given that they made their way into ConceptNet at one time as plausible assertions.

As a by-product of the annotation task, we also obtained an evaluation of the truthfulness of the assertions contained in ConceptNet, and the results on this evaluation are given in Subsection 3.5.5.

RELATION	MEANINGFUL	TOTAL
AtLocation	148	452
IsA	150	477
HasProperty	158	516
MotivatedByGoal	151	450
SymbolOf	80	156

Table 5: Meaningful and meaningless assertion decomposition of ConceptNet-based training datasets.

3.5.2 *Candidate assertion mining*

BagPack models rank novel assertions that are constructed by combining pairs of concepts frequently co-occurring in Wikipedia.

Unlike in the SAT or selectional preference tasks, where we are given a list of concept pairs in advance, to extract commonsense assertions from free text, we need a way to harvest *candidate* assertions, that can then be ranked by the algorithm.

The candidate assertions are mined from the syntactically parsed Wikipedia corpus made available by the WaCky project (see link above). The top 10,000 most frequent verbs, nouns and adjectives were considered as potential concept heads, and I extracted potential concept phrases with a simple grammar aimed at spotting (the content words of) noun, verb and adjec-

tive phrases (for example, the grammar accepts structures like `Adj Noun`, `Verb Adj Noun` and `Adv Adj`). In this phase, I was not interested in the semantic association between the concept pairs but simply tried to generate lots of pairs to feed to the trained BagPack models.

The pair extraction algorithm applied to Wikipedia produced 116,382 concept pairs. Then, I randomly sampled 5,000 pairs (containing 5,385 unique concept phrases) from this set, and generated 10,000 directed pairs by ordering them in both directions. Approximately 68% of the concepts in the sampled pairs were single words, 30% were 2-word phrases, 2% contained 3 or more words. Some example concept phrases that were mined are *wing*, *sport team*, *fairy tale*, *receive bachelor degree*, *father's death*, and *score goal national team*.

Finally, I associated the sample pairs with each of the five relations I study, obtaining a set of assertions that contain the same concept pairs, but linked by different relations. This step resulted in 10,000 candidate assertions for each relation.

3.5.3 *Gold standard for common sense*

In order to provide a gold standard for further analysis, two expert raters annotated approximately 400 assertions for each relation. This sample was picked post hoc, consisting of the candidate assertions that were ranked top by the BagPack models among the initial 10,000. The raters' overall Cohen's kappa was 0.37. The raters agreed on 183 meaningful assertions (8.8%) and 1,508 meaningless assertions (72.6%). Any assertion that was annotated as meaningful by at least one rater was assumed to be meaningful for purposes of assessing performance. As a side note, the observed Cohen's kappa is very low compared to other tasks reported in the literature. However, in commonsense data annotation it is very hard to achieve higher agreement ratios. For example, in Schubert and Tong [2003], the reported mean kappa between pairs of raters on a 3-way decision task (involving categories "true", "false", and "undecidable") is 0.375.

3.5.4 *Experimental setup*

SOURCE CORPORA AND MODEL IMPLEMENTATION For each of the five datasets coming from the previous steps, I trained a separate BagPack model. I extracted the co-occurrence vectors from the Web-derived English Wikipedia and ukWaC corpora as I did for the SAT analogy task.

Since ConceptNet concepts are often expressed by multiple words (*Sacher Torte*, *eat too much*, ...), I employed a shallow search

for the concept phrases. Basically, for a single phrase, I looked for the occurrence of the constituents with possible intermittent extra elements. We say a concept *occurs* in a sentence if all its constituents occur in the same order in the sentence with possible intermittent extra elements, and if the concept phrase spans no more than twice of its original length (e.g., if the concept is a 4-word phrase, it can span at most an 8-word range). Two concepts are said to be *co-occurring* if both concepts occur in the same sentence, they do not overlap (i.e., the last word of the first concept comes before the first word of the second concept), and the range that both concepts span together is not longer than 20 words (i.e., at most 18 elements can occur between the first word of the first concept and the last word of the last concept). For efficiency reasons, a maximum of 1,000 sentences were used to extract co-occurrence statistics for a given pair. The features in the co-occurrence matrix were weighted by PMI. The features were restricted to the most frequent 5,000 lemmas in ukWaC, resulting in a 15,000-dimensional vector for each pair.

Following the suggestion of Hsu and Chang (2003), before SVM training, each feature t's $[\hat{\mu}_t - 2\hat{\sigma}_t, \hat{\mu}_t + 2\hat{\sigma}_t]$ interval is scaled to $[0, 1]$, trimming the exceeding values from upper and lower bounds (the symbols $\hat{\mu}_t$ and $\hat{\sigma}_t$ denote the average and standard deviation of the feature values respectively). I use the C-SVM classifier as implemented in the Matlab toolbox of Canu et al. 2005 with a linear kernel and the cost parameter C set to 1.

3.5.5 *Results*

EVALUATION OF CONCEPTNET A by-product of the annotation stage of the training set is an evaluation of the quality of the assertions in ConceptNet. Remember that the training set contains two kinds of assertions, the original assertions based on ConceptNet, which are likely to be true, and the artificial assertions, which are randomly crafted and are likely to be false. If we limit ourselves only to the responses given for the *original* assertions in the training set, we can get an estimate for the precision of ConceptNet. In a similar fashion, the *artificially generated* random assertions can provide a baseline for sanity check. In Table 6, we see the ratio of the number of assertions annotated as meaningful to the total number of assertions (i.e., precision) for each relation, separately for the original and artificial subsets. The overall precision of the sample from ConceptNet is calculated to be around 0.65. This estimation is in accordance with the previous estimations of the precision of ConceptNet [Havasi et al., 2009].

RELATION	PRECISION (ORIGINALS)	PRECISION (ARTIFICIAL)
AtLocation	0.68	0.15
IsA	0.63	0.17
HasProperty	0.59	0.23
MotivatedByGoal	0.69	0.15
SymbolOf	0.69	0.09

Table 6: Precision measures for the original and artificially generated assertions based on ConceptNet.

EVALUATION OF BAGPACK As performance measure, I report the areas under the ROC curves (AUC). The area under the ROC curve can be interpreted as the probability of a random positive instance having a higher confidence score than a random negative instance [Fawcett, 2006]. An AUC value of 0.5 implies chance performance. Using AUC allows us to focus on the discriminative power of a model without having to pick a threshold.

Table 7 reports the AUC obtained by the BagPack models trained on ConceptNet-based training sets and evaluated on the Wikipedia-based test sets (the gold standard candidate assertions). For all relations, the performance of BagPack is significantly above the random baseline. However, AUC for MotivatedByGoal is barely above chance level and even the best AUC performance of 0.66 that is obtained on AtLocation is quite low, suggesting that BagPack alone cannot be used to extract reliable commonsense assertions from corpora.

RELATION	AUC
AtLocation	0.66
IsA	0.58
HasProperty	0.59
MotivatedByGoal	0.59
SymbolOf	0.58

Table 7: BagPack AUC on five ConceptNet relations.

3.6 CONCLUSION

Although the results I presented for commonsense collection are above random baselines, they are far from being perfect and are parallel to what have been reported in the literature with similar approaches [Yu and Chen, 2010]. However, when BagPack is evaluated on arguably more constrained semantic tasks of analogy and selectional preference, its performance is comparable to the state of the art.

Learning commonsense facts from unstructured text is an *AI-hard* task in its most general definition – one cannot expect to solve the commonsense problem before solving the AI problem. My formulation of the task as extending the ConceptNet is also a very general task as it contains a variety of relations and the test set is picked directly from corpus without any filtering. Therefore, it should not be surprising to see performance with relatively simple algorithms. Therefore, I conclude that if we want to substantially improve the accuracy in commonsense extraction

without a trade off in recall, we should employ qualitatively different approaches. It is precisely this observation that motivates the human-computational approaches which tap ordinary people's knowledge and effort in order to collect common sense. In the next chapter, we will see an example of such an attempt.

4 CONCEPT GAME

Jerry: Oh, volunteer work! [...] People being helped by people other than me. That makes me feel good inside.

— Seinfeld, The Pick

Human-based computation is a workaround technique employed in computer science to tackle with problems that are AI-hard – by outsourcing the *hard* sub-steps of the problem to people [Von Ahn et al., 2006]. Some human-based computation methods employ *games with a purpose* – which are designed specially to tap people's efforts while the people play a game and have fun. In this chapter, I present such a method that aims to create a commonsense repository by letting people play a Web-based game.

Human-computation depends on the idea of recruiting people to solve computational problems that are not easily solvable by current techniques in computer science. Games with a purpose pose human-computation problems as fun-to-play games.

4.1 RELATED WORK

Quinn and Bederson (2009) classify different approaches to harnessing the computation capabilities of humans [Quinn and Bederson, 2009]. Among the several dimensions they use for categorization, the following are relevant for us:

- Motivation of users
- Techniques for coping with noise
- Minimum participation time
- Cognitive load placed on the player.

Below, I discuss several human-computation-based approaches to commonsense knowledge collection along these dimensions. A summary of this discussion is given in Table 8.

Verbosity[1] is a word-guessing game very much like the commercial game "Taboo" [Von Ahn et al., 2006]. In a single round of the game, a describer tries to tell the secret word to a guesser by filling in one or more of the slot-based templates he is given. The templates are crafted in order to collect commonsense knowledge about the secret word. Some examples are "It is a type of ___", "About the same size as ___". If the guesser is able to find out the secret word, the clues that the describer provided are considered to be true and stored.

1 http://www.gwap.com/

GAME	NOISE-FILTERING	MINIMUM PARTICIPATION	COGNITIVE LOAD	SOURCE OF COMMON SENSE
FACTory[1]	Majority	Seconds	Low	Cyc
Verbosity[2]	Extensive filtering	Tens of seconds	High	Players
Common Consensus[3]	Majority	Seconds	High	Players and OMCS
Concept Game	Majority	Seconds	Low	Corpora

Table 8: Summary of GWAP for Commonsense Collection. [1] http://game.cyc.com/; [2]Speer et al. [2010]; [3]Lieberman et al. [2007].

Common Consensus [Lieberman et al., 2007] is another game with a purpose based on the popular TV show "Family Feud" where the player is asked to guess the popular answers for a given question like "Something that is likely to be found in a kitchen is ___?". The more popular an answer is the more points are obtained; thus, players try to guess the most *commonsense* answers.

Both Verbosity and Common Consensus are used to populate facts in OMCS and consequently in ConceptNet [Speer et al., 2010]. Both are games with a purpose where the players' primary motivation is having fun and commonsense knowledge collection is a side effect of the game playing. Thus, they differ from the volunteer-based knowledge collection effort of OMCS. In order to cope with noisy input, both games can employ redundancy and accept a fact as true only after it is asserted by more than one player. However, the players are free to introduce the facts they like and that means many valid facts may be asserted only once – resulting in a sparse dataset. The games can only guide the players to provide information about a given domain or a target concept but they lack the ability to validate a given commonsense fact. Coupled with the tendency of players to abuse the system, especially for Verbosity where the players try to do their "best" to convey the secret word, noise reduction and validation call for non-trivial and usually heuristic-based approaches [Speer et al., 2010]. For both games, the overhead of playing is quite low and players can almost immediately start to contribute. However, the burden of producing commonsense facts is placed on the players. In Verbosity, it is up to the describer to think and find good clues. In Common Consensus, the player has to come up with plausible candidates for the popular answers. Thus, the games are cognitively engaging and the cognitive load on the player is quite high.

Another application, which is related to Cyc, is the FACTory game[2]. In FACtory, the player is shown a set of commonsense statements and is asked to express his opinion about the statements. The FACTory's commonsense statements are generated from the CYC repository, and players must tell whether they think the statements are true or false. Extra points are awarded when a player agrees with the majority answer for a fact and a certain consensus threshold has been reached. Although the system is presented as a game – and thus, the motivation of answering the questions is expected to be having fun – in its current stage, there is no clear fun aspect in FACTory – which rather looks like an interactive and marginally less boring mechanism to collect human opinions about a given set of commonsense

2 http://game.cyc.com/

statements. An important difference between FACTory and the previous two games is that FACTory relies on Cyc as a source of commonsense statements. Therefore, all it has to do is to ask for the opinion of the player. This has the benefit of reducing the cognitive load on the player (it is easier to answer yes to the question "Can salmon be found in a fridge?" rather than to try to come up with "salmon" as an answer to the question "What can be found in a fridge?".

There are other attempts to crowdsource commonsense data evaluation/collection, by utilizing services like Amazon's Mechanical Turk[3], e.g., Gordon et al. [2010a]. I consider such crowdsourcing methods as complementing the games-with-a-purpose approach rather than competing with it. For the evaluation of small datasets, crowdsourcing by payment may be a convenient alternative, but as the amount of data to be annotated increases so does the cost of annotation. In contrast, the operational costs of games are usually almost constant (e.g., a small monthly reward to motivate players) and enlarging the user base would not incur any additional costs. Current estimates are that the unit cost per response for Concept Game in its initial phase has been comparable to, if not less than, the cost reported for Mechanical Turk services (around 300 to 500 responses per USD spent) [Gordon et al., 2010a]. As (read if) the game becomes more popular among Facebook users, the unit cost will get much lower.

4.2 THE GAME

Concept Game is open to public as a Facebook application.

Concept Game[4] (CG) is a game with the purpose to collect common sense from laypeople. It is based on the idea that production of verbal information is a significant burden on the player and, thanks to the text mining base, it is possible to design enjoyable games that do not require the players to produce assertions. Therefore, the game aims to achieve its purpose not by having the players produce commonsense assertions, but having them verify already collected candidate assertions. This approach allows us to design fast-paced games where the interaction between the user and the game is limited to an expression of assent. CG is presented in the context of a slot machine which produces random assertions. A meaningful assertion is a winning configuration. The trick is that the winning configurations do not dispense rewards automatically, but first they have to be recognized by the player to "claim his points". In this way, players tell us which assertions they find meaningful.

3 http://www.mturk.com/
4 http://apps.facebook.com/conceptgame/

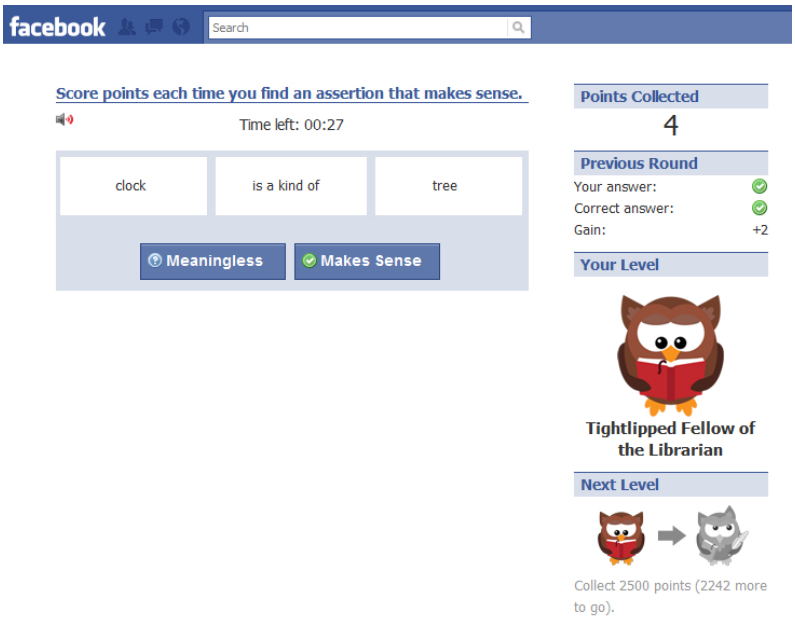

Figure 3: Screenshot of a playing session in Concept Game

The game consists of independent play sessions each of which starts with an allocation of 40 seconds. First, the player sees three slots with images of rolling reels. They correspond to an assertion's left concept, relation, and right concept. Then, the contents of the slots are fixed one by one with some values picked from the database and as a result an assertion is displayed. At that point, the player has to press one of two buttons labeled as "Meaningless" or "Meaningful". If the player presses the meaningful button this can result in two different outcomes: either the displayed assertion is indeed meaningful and he is rewarded with two points and two bonus seconds (i.e., true positives are rewarded), or the assertion is in fact meaningless and the player loses three points and three seconds (i.e., false positives are penalized). However, pressing the meaningless button does not change the score or the remaining time (i.e., neither false negatives are penalized nor true negatives are rewarded). The feedback is conveyed to the player visually and acoustically (e.g., in case of a reward a green color flashes, in case of a penalty a red color flashes). The reels roll again, and the process repeats. This continues until the end of the allocated time, which can get longer or shorter depending on rewards and penalties. A typical screenshot of the game is given in Figure 3.

In the previous description, I pretended that the game already knows which assertions are meaningful, and rewards or penalizes the user accordingly. This, of course, is not the case (or else, the game would be useless). In CG, candidate assertions (i.e. assertions whose labels are unknown) that are produced by the text

miner are shown to the players while using a validation policy similar to the honor-based, proof-of-payment method employed in many public transportation systems. In such a system, instead of checking every response, periodic controls are carried out to make sure the abuse of the system is effectively discouraged. In the current implementation, the probabilities of showing a known meaningless, a known meaningful, and a candidate assertion are 0.4, 0.3, and 0.3 respectively. In other words, 30% of the collected responses are for the new candidate assertions proposed by BagPack. For the candidate assertions, whatever the player responds is accepted as the correct answer and this is the actual knowledge we want to harvest. The meaningless assertions are used to verify that the player is not abusing the game. The meaningful assertions are used to make sure the player scores points and does not get frustrated. Note that increasing the precision of candidate generation would help us to display more unknown assertions without a significant impact on the game experience (i.e. players would still be able to score points without the support of the pre-selected meaningful assertions). In addition, the set of known meaningful assertions expand as the players continue to play and validate the meaningful candidates. To allow the players to warm up, I implicitly train them and do not show any unknown assertions until they complete three sessions with positive scores.

Technically, the game is almost equivalent to asking a group of raters to tick those assertions from a list which they think make sense. This is a dull task especially if there are few meaningful assertions compared to meaningless ones. In the context of a slot machine, however, the experience of seeing many meaningless assertions becomes part of the game, which creates an expectation in the player that is – hopefully – resolved with a "winning" configuration. The relatively short session timing, combined with the need to be accurate because wrong claims are penalized, should keep the attention level of the players up, and consequently add to the fun. I made sure that players are aware of their achievements (they see total and session scores they have collected) and have an incentive to keep playing. There is also a top score list that shows the users who scored highest in a single session. I implemented a ladder system where the players are represented by cute avatars. Taking advantage of the integration with Facebook, I ask players' permission to post their activity in the game to their public walls and give them the opportunity to invite their friends in order to go up in the ladder.

At the time this manuscript was written, 690 Facebook users had tried the application. Out of them, 146 passed the implicit training session and actually contributed to my dataset. After the first visit to the application page, more than half of the

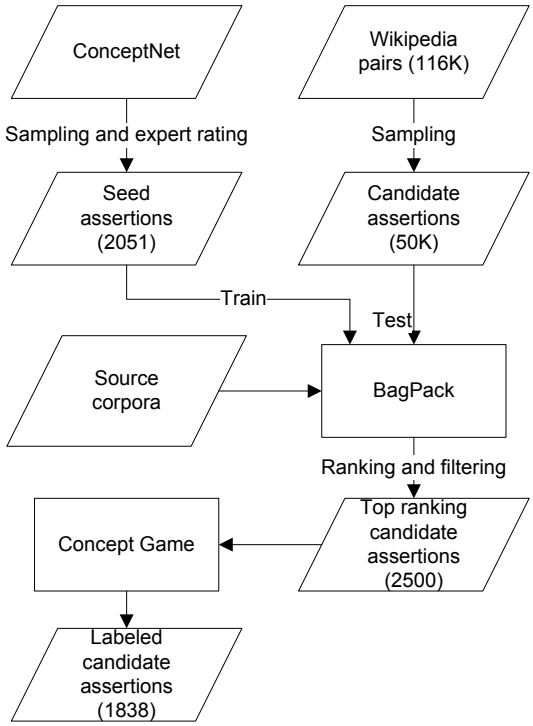

Figure 4: Data flow for the kick-starting experiment. Concept pairs are extracted from Wikipedia as explained in Section 3.5.1. BagPack uses Wikipedia and ukWaC corpora to extract vector representations of assertions. Numbers in parentheses are the number of assertions in the datasets.

players (58%) returned and played the game at least on one other day while more than one fourth (28%) returned at least on two other days. During a two-month period, I collected over 110,000 responses, approximately 25,000 of them for unknown, potentially meaningful assertions.

4.3 KICK-STARTING

The aim of Concept Game is to take a set of candidate assertions and have them filtered by people who play the game. In our case, the output of BagPack provides us such a set of candidate assertions. In the next subsection, we will study an experiment that is aimed to evaluate the efficiency of Concept Game in separating the meaningful assertions from the meaningless ones by using the players' responses.

Concept Game acts as a filter – that passes only the assertions found meaningful by the players.

4.3.1 *Experimental setup*

The experimental procedure is summarized in Figure 4. The seed assertions based on ConceptNet and manually checked as described in Section 3.5.1 serve as the BagPack training data, and

the set of assertions mined from Wikipedia (see Section 3.5.2) serve as candidates to be ranked by the Concept Game. As already explained in Section 3.5.5, a separate BagPack model was trained for each relation and the candidates were ranked according to the confidence scores of these models. For each relation, I kept the top 400 ranking assertions as the set to be fed into the game. I will compare the performance of the labeling obtained from the game to the results I obtained by the "pure BagPack" approach in Section 3.5.5. Note that the gold standard for the test is obtained by the two expert raters annotation described in Section 3.5.3.

Once BagPack ranks and filters candidate assertions, they are ready to be fed into Concept Game. For this purpose, 18 people were contacted by email and were invited to play the game, mostly college students and staff that the author personally knew. Remember that in previous steps the raters were experts, but this time the players are recruited from *non*-expert people. The game was open to this "semi-public" for approximately 10 days. I used the negative training assertions for control purposes to penalize players' wrong decisions.

4.3.2 *Results*

In total, 25 players (7 presumably invited by the ones I contacted) responded and provided a total of 5,154 responses for the candidate assertions. The ratio of players who scored an assertion as meaningful is the *CG score* of the assertion. In addition to CG scores, BagPack confidence scores are used as the *BagPack scores*.

During the analysis, I considered the 1,838 assertions which received at least two meaningful or two meaningless responses, split across relations as shown in Table 9. The assertions that were labeled as meaningful consisted of 547 unique concepts and 86% of the them were not attested in the ConceptNet-based dataset (23% were not attested in the entire ConceptNet knowledge base).

RELATION	MEANINGFUL	TOTAL
AtLocation	71	385
IsA	63	328
HasProperty	95	362
MotivatedByGoal	65	408
SymbolOf	73	355

Table 9: Summary of the candidate assertion sets. Labels are decided by majority votes based on CG players' responses with ties broken in favor of being meaningless.

RELATION	CG	BAGPACK
AtLocation	0.76	0.66
IsA	0.79	0.58
HasProperty	0.69	0.59
MotivatedByGoal	0.72	0.59
SymbolOf	0.71	0.58

Table 10: Area under the ROC curve (AUC) on candidate assertion set.

Using the expert raters' judgments as the gold standard for the candidate assertions, I computed relation-specific ROC curves for the CG. The areas under the ROC curves are given in Table 10 – I repeat the BagPack's AUC values for easier comparison. As an illustration, the ROC curves obtained for AtLocation are given in Fig. 5.

We observe that when the top BagPack candidate assertions are ranked by using the answers of the players, the performance considerably increases for all relations. This proves two points: First, BagPack alone is not sufficient to evaluate candidate assertions mined from Wikipedia reliably, and second, Concept Game is able to improve performance. As an ad hoc evaluation, I computed the Cohen's kappa between the gold standard of the raters and the output of Concept Game. The kappa value was 0.39 (comparable to the kappa value of 0.37 between the two raters themselves). Coupled with the high AUC values reported

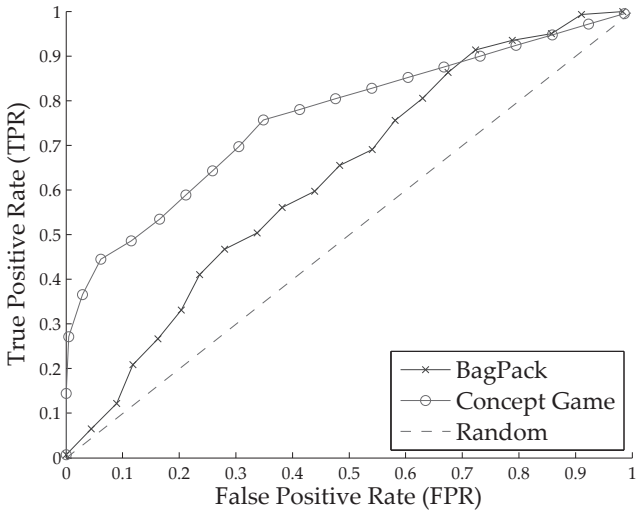

Figure 5: ROC curves for AtLocation.

for the CG scores, I conclude that the annotation of the game players is comparable to manual annotation by experts.

4.4 BOOTSTRAPPING

In the previous experiment, we saw that combining BagPack and Concept Game can lead to a high-quality dataset of commonsense knowledge. So far, the criterion for quality has been the judgments of two expert raters. Now, I want to show that the output of the combined architecture can actually improve the performance in a more realistic task. Bootstrapping the entire system with its own output provides us with such a task. By bootstrapping, I mean training a new BagPack model only on the assertions mined from Wikipedia – with their labels decided by the Concept Game players. This experiment can be thought of as a continuation of the previous experiment where I use the labeled candidate assertions (the output of kick-starting) as the training seed of a new round of BagPack training, like adding a "train arrow" from the box at the bottom of Figure 4 to the BagPack box. I compare the performance of the bootstrapped BagPack and the original (i.e., ConceptNet-based) BagPack on a new evaluation set. That will allow us to see if the output of Concept Game is of sufficient quality to allow such a bootstrapping.

4.4.1 *Experimental setup*

In this experiment, I employ two different BagPack training seed assertions sets: The ConceptNet-based assertions and the candidate assertions annotated by Concept Game in the previous experiment. For obvious reasons, the latter is called the *bootstrap* dataset from now on. In addition, I combined the two into a third dataset, the *combined* dataset. Descriptive statistics for the ConceptNet-based and bootstrap training sets were already given in tables 5 and 9, respectively – the former as a result of expert annotation and the latter as a result of game playing.

To construct a final evaluation dataset (which will be called the *evaluation* dataset from now on), I randomly sampled approximately 1000 assertions for each relation by using the pairs mined from Wikipedia. I did not pick a set of candidates ranked by BagPack, as I did in the previous experiments, but used a random sample of assertions because I wanted to compare the performance of different BagPack models on the same evaluation set – ranking and filtering the assertions by any of the BagPack models would create a bias. I made sure, moreover, that, for each relation, the three corresponding BagPack seed datasets are disjoint with the evaluation set (i.e. they do not have any common

RELATION	MEANINGFUL	TOTAL
AtLocation	120	985
IsA	39	978
HasProperty	67	983
MotivatedByGoal	51	980
SymbolOf	41	973

Table 11: Meaningful and meaningless assertion decomposition of evaluation dataset.

assertions). The evaluation assertions were rated by the players of CG during a two-month period between April and May 2010 and they received at least two responses from different players. I already gave some details on this game playing session at the end of Section 4.2. For the annotation of the evaluation dataset, majority vote was used with ties broken in favor of being meaningless. The details of the evaluation dataset are given in Table 11.

4.4.2 *Results*

In Figure 6, I report the area under the curve (AUC) values of the three BagPack models for each relation. The error bars represent the confidence intervals at 95% significance level obtained by 5000 resamples with replacement.

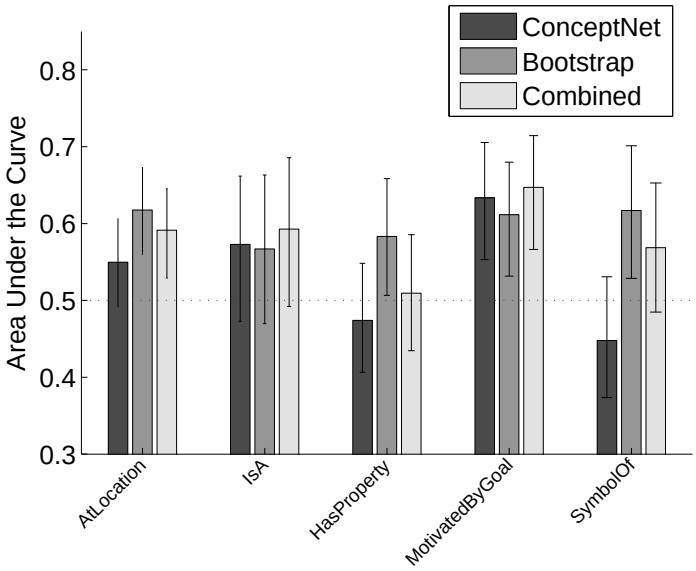

Figure 6: AUC values for BagPack models trained on different datasets, for all relations. The dashed line at $y = 0.5$ is the chance-level performance. The error bars span the 95% confidence intervals. The gold standard is based on Concept Game players.

For AtLocation, HasProperty, and SymbolOf relations, bootstrapping obtains even better results compared to using original ConceptNet-based assertions. For IsA and MotivatedByGoal, bootstrapping is almost as good as using the original training dataset. The combination of two datasets brings additional improvements in IsA and MotivatedByGoal but the differences are not significant.

It is possible to use the output of the combined system as the input of a second round of text mining without observing a decrease in performance.

Note that in this experiment, the gold standard for the evaluation dataset is provided by Concept Game, not by the experts, as the first experiment already showed that the output of Concept Game is of sufficient quality. The fact that both the bootstrap and evaluation datasets are annotated by the same process (i.e. Concept Game) may create an advantage for bootstrapping. Since the actual goal in this task is to find candidate assertions which are likely to be found meaningful by the players, I see this effect of shared annotation mechanism as an opportunity to be seized, not an external variable to be controlled. Nevertheless, to see the extent of the effect, I manually annotated the entire evaluation dataset and I replicated the experiment with this ratings as the gold standard. The performance of the bootstrapped BagPack is not significantly worse than the original BagPack – its AUC values for the AtLocation, IsA, and HasProperty relations are higher than the original BagPack – though the differences are much less pronounced.

Another possible confounding factor I considered is the amount of overlap between the datasets in terms of concepts. The bootstrap dataset and the evaluation dataset come from the same population of assertions that are mined from Wikipedia. Therefore, even though they are disjoint in terms of assertions, they have a significant number of common concepts and that may be one of the reasons why I obtain better results by bootstrapping. On the average, 21% of the unique concepts in the bootstrap dataset also occur in the evaluation dataset. In comparison, approximately 16% of the unique concepts in the ConceptNet-based dataset occur in the evaluation dataset. In order to control for this shared-concepts effect, I marked all concepts which are part of positive instances in the evaluation dataset and removed all assertions which contain these concepts from both training datasets. As a result, the total number of assertions in the bootstrap dataset reduced from 1838 to 1646 and the number of assertions in the ConceptNet-based dataset reduced from 2051 to 1892. Predictably, almost all computed AUC values are lower compared to the previous experiments with the untouched datasets. However, even though the decrease in AUC for the bootstrap dataset is visibly higher, bootstrapping results are almost as good as the original BagPack results. The fact that the gain of bootstrapping dimin-

ishes when I remove the common concepts from the training and evaluation sets is not discouraging because in real-world settings the bootstrap datasets will come from the same population of the future evaluation sets (as it does in our case), therefore a certain amount overlap is what is to be expected.

Summarizing the second experiment, the bootstrap dataset – which was mined from Wikipedia, filtered by BagPack and annotated by the 25 Concept Game players – is at least as successful as the ConceptNet-based dataset – which depends on ConceptNet and was annotated by 22 experts – in seeding BagPack to extract commonsense knowledge from corpora.

4.5 CONCLUSION

Concept Game is implemented as a fully public Facebook application that lets laypeople validate a set of candidate assertions while they play a game. Currently, the source for candidate assertions is a text mining process (i.e. BagPack), but in principle any – possibly noisy – source of commonsense knowledge would do the job.

As a game with a purpose, Concept Game has several distinctive aspects compared to other games reported in the literature. First of all, it does not rely on the players for generation of common sense but on the text corpora; thus, it is able to tap domains that players do not talk about when they are asked directly. The verification-based setting allows a fast-paced game and helps us to explore different areas of the game-design space – possibly allowing us to recruit different types of players.

Concept Game is a fully functional and public Facebook application. In the short term, I am looking for ways to make the game more attractive to a wider non-specialized audience. I would like to convert the lemma sequences produced by BagPack into natural sounding sentences. I have recently started to offer small gifts to top players as an incentive to start and keep playing.

A recent (and raw) snapshot of the data collected is downloadable from the Web[5]. Once it gains a reasonably wide player-base and construct a larger dataset of commonsense assertions I plan to share the dataset in a more structured form.

In future analyses, I would also like to look for cultural differences in assertions that receive contrasting ratings from players from different continents. Using Facebook as a platform allows us to access demographics of players for statistical analysis.

As a final remark, we first saw that when supplied with a noisy set of candidate assertions (i.e. Wikipedia pairs), Concept Game is able to separate the meaningful and meaningless assertions with

5 http://github.com/amacinho/Concept-Game-Datasets

an improved performance compared to BagPack alone thanks to the responses of players. More importantly, as a semi-external task of validation, we saw that the output of the game can be used as an effective training dataset for a second round of BagPack-based text mining. This is a strong evidence that the combined architecture of BagPack and Concept Game is able to extend the starting knowledge base without sacrificing precision.

5

STEREOTYPICAL KNOWLEDGE AND COMMON SENSE

Common sense is the collection of prejudices acquired by age eighteen.

— Albert Einstein

Lisa: Dad, women won't like being shot in the face.
Homer: Women will like what I tell them to like.

— The Simpsons

In the previous chapters, I presented two rather complementary methods of collecting common sense, text mining and human computation. Now, I move on to a more exploratory topic and pursue the idea of teaching stereotypical beliefs and prejudices to the computers.

In this chapter, our interest will be on extending the domain of machine-accessible commonsense knowledge. Without an explicit knowledge of the stereotypes, these beliefs will be implicit, hidden and intermixed with other "objective" facts in a knowledge base. To quote Sherron (2000) – who talked about Cyc in particular – "Cyc's common sense might very well 'believe' certain stereotypical ideas about women, gender, sexual orientation, etc., and then make inferences based on those 'beliefs'. Without a strong challenge a homogenizing effect occurs, solidifying the original stereotype among users of the program." [Sherron, 2000]. I believe my work may help to make such knowledge explicit, thus enable us to deal with the problem of stereotypical beliefs.

I focus on stereotypical gender expectations about actions (i.e., verb phrases) as a case study. Though, the techniques I propose can work for any kind of phrase. ConceptNet/OMCS already contains a substantial number of verb phrases which correspond to actions that humans carry out in a daily manner. Consequently, I use ConceptNet as a source of the actions that are pertinent to the daily life of people. Apart from the semantic relations between these actions that are provided by ConceptNet (e.g., *motivated by*, *caused by*, *has first subevent*), the stereotypical expectations about the actions should also be an important part of our commonsense repository: whether it's right or wrong, an AI should know that (we expect that) *women like shopping* and *men like football*. I propose a methodology that will allow us to extract *gender features* from large amounts of text and from the metadata provided by social media (in the specific case, Twitter). Subsequently, the

It is common sense to know that people regard becoming a nurse as a feminine action, while using a hammer as a masculine one – regardless of whether it is true or not.

gender features are used to tag the actions in the ConceptNet repository as *masculine*, *feminine*, or *gender-neutral*.

In a nutshell, my strategy is to extract the gender features for a given verb phrase from corpora, compute the *gender bias* of the phrase which quantifies whether it is a masculine or feminine action, and compare the predicted bias to the stereotypical expectations of people, collected in a rating task. I focus on two methods. The first one relies on the metadata-guessed gender of Twitter users and their tweets. The second employs a heuristic based on the gendered pronouns and names referred to the actions observed in a large corpus of Web documents.

5.1 RELATED WORK

In her influential paper of 1973, Robin Lakoff characterizes some distinguishing aspects of women's speech as being about trivial issues, apologetic, and being non-assertive [Lakoff, 1973]. In making such claims, her main source of data is introspection and anecdotal observation. Although her approach has been influential and instrumental on studying the gender differences of language use, later data-driven studies have empirically contested some of her claims [Holmes, 1990].

More recently, increased access to larger volumes of textual data made it possible to search for (and of course find) more subtle differences in the language use of the two genders. Such an example is the work of Argamon et al. [2003] which studies the gender differences in formal written texts consisting of samples from the British National Corpus. Some of their key findings are that females use many more pronouns than males, and males use many more noun modifiers than females.

Another source of textual data for corpus-based studies on gender differences is the Web blogs where people contribute content in a less formal way. It is possible to collect metadata like the gender of the posters, create two sub-corpora based on the text created by the two genders respectively, and then compare the two sub-corpora to reveal statistical differences between the linguistic patterns or content [Argamon et al., 2007]. Such an approach allows to tap a larger amount of text thanks to the millions blogging about virtually everything on a regular basis. The work of Argamon et al. [2007] confirms the gender differences previously found in formal texts, and in addition, provides evidence that there are substantial differences in the content as well (e.g., females are blogging about past actions more frequently than males and males are blogging about politics more frequently than females).

Another work on data-driven gender modeling is the study of Liu and Mihalcea [2007] which is also based on text analysis of blogs. This study is particularly relevant for us because it is based on the gender preferences for several dimensions that are pertinent to user-interface designs (e.g., gender preferences of foods or color and size of things) which happen to be also salient dimensions for commonsense knowledge [Liu and Mihalcea, 2007].

In this study, my approach is similar to the work of Liu and Mihalcea [2007] in the sense that I am not interested in the gender differences in language use, but rather the differences in the actions that the two genders are associated with. In this case, language acts as a proxy to extract such differences.

5.2 MATERIALS

5.2.1 *Corpora*

The first corpus I worked on is the Edinburgh Twitter Corpus (ETC) already explained in Section 2.2. I want to utilize the gender information of the users, but the corpus does not originally contain the metadata about the users who posted the tweets. In order to get the metadata, I queried the Twitter API to obtain the name of each user. Then, I used the first names of the users to guess their genders – as Twitter does not disclose the gender of the users via its API. I used two lists of the most popular American male and female names – compiled from the public data provided by the US Census Bureau and US Social Security Administration (the lists are made available at this URL: `http://github.com/amacinho/Name-Gender-Guesser`). Any user whose first name was not on one of the two lists was discarded from further consideration. Finally, I separated the tweets into two sets according to the *guessed* gender of the users and obtained two sub-corpora containing 82 million tokens for males and 89 million tokens for females (5.2 million male tweets and 5.9 female tweets), each contributed by approximately one million users.

Utilizing the first names of people to guess their gender is inherently a noisy (and possibly biased) process because of several reasons including bogus names provided by the users and unisex names. Nonetheless, given the lack of true gender information, first name is a very strong (maybe the strongest) clue of the gender.

Unfortunately the lack of a gold standard of the genders of users also avoids a thorough evaluation of the accuracy of the guessing method. Nevertheless, it is possible to carry out some sanity checks to make sure the results are in accordance with what

we expect. In Table 12, we can see the distribution of male and female users who mention the phrases *my husband*, *my boyfriend*, *my wife*, and *my girlfriend* at least once in their tweets. We can see that the phrases relating to the male significant others or partners are mentioned much more frequently by females and vice versa. Obviously, this is not a direct proof but an evidence suggesting the gender guessing method is reasonably accurate.

PHRASE	MALES (%)	FEMALES (%)
my husband	7	93
my boyfriend	10	90
my wife	87	13
my girlfriend	73	27

Table 12: The frequency of male and female users who mention specific phrases (based on unique user count). The frequencies are computed on a balanced sample of Twitter users that have equal number of male and female users.

Employing a corpus that is based on social media with the meta data about the contributing users (i.e., Twitter) allows us to keep track of who says what. Employing a larger Web-based corpus (i.e., ukWaC) allows us to keep track who is reported to be doing what.

In addition to the Twitter data, I also used ukWaC, the 2-billion-token corpus introduced in Section 2.2 [Baroni et al., 2009], and experimented with the widely used 100-million-token British National Corpus[1], but the latter's coverage of ConceptNet phrases was so low that I did not pursue this option further. Compared to ETC, ukWaC is a more traditional corpus made of relatively long documents, and it does not provide metadata about who uttered/wrote the collected text and when. Consequently, I employed linguistic heuristics to extract gender features as we will see in Section 5.3.

5.2.2 *Common sense actions*

ConceptNet – the commonsense semantic network that is based on OMCS – which was introduced in Section 2.1 serves as the source of commonsense actions. The total number of unique concepts in ConceptNet is 267,364 – most of which are multi-word phrases (e.g., *door knob*, *apple tree*). From this larger set, I picked a subset of "actions", i.e., concepts corresponding to verb phrases. In this context, a verb phrase is simply a multi-word expression beginning with a verb. In turn, a verb is any word that is tagged much more frequently as a verb than any other part-of-speech in the ukWaC corpus. The number of actions obtained from ConceptNet this way is 49,754. As we will observe, ConceptNet also contains some spurious or meaningless concepts

1 http://www.natcorp.ox.ac.uk/

like "'do what' in the assertion *(preserve, AtLocation, do what)*. I did not attempt to filter these concepts/actions.

5.2.3 *Gold standard*

As the gold standard dataset, I randomly sampled 702 phrases from the set of actions detected in the ETC Twitter corpus and represented in ConceptNet. I give the details of phrase detection in Section 5.3. I employed Crowdflower's[2] crowd-sourcing services to have the verb phrases annotated by people. I used a 5-point scale in the annotation task: Typically feminine (-2), slightly feminine (-1), neutral (0), slightly masculine (1), and typically masculine (2), with an extra option of "Not a verb phrase"/"meaningless". Each rater was presented a verb phrase and was asked to provide his/her opinion about the phrase. After the data collection phase, I eliminated the phrases whose majority answer was "meaningless" or which did not receive at least five responses. This filtering results in 441 phrases. Overall, 112 raters contributed (each rater annotated at least 18 verb phrases) and for the 5-choice question, the average agreement rate of each rater with the majority answer for a phrase was 62% (including the phrases that are eliminated as meaningless). I did not keep track of the gender of the raters.

For each verb phrase in the dataset, I calculated the *gender score* as the mean score of the responses it received. This score serves as the *human* gold standard for the stereotypical gender expectation of the corresponding action. For illustrative purposes, I provide a stratified random sample of the gold standard dataset in Table 13.

5.3 CORPUS ANALYSIS METHODOLOGY

I searched for the ConceptNet phrases (both actions and non-actions) in the lemmatized version of the corpora, allowing at most one intermittent token between two lemmas of the phrase (e.g., the concept *long holiday* is said to be *observed* when we encounter the text "long holidays" or "longing for holiday").

In the case of ETC, in order to compute the gender bias of an action, I detect all of its corresponding verb phrase's utterances in the male and female sub-corpora and compute the proportion of the number of male utterances as the raw scores. I report normalized versions of the raw scores, computed using expectation and variance values from a binomial distribution with p equal to the overall proportion of male utterances and n equal to the

2 http://www.crowdflower.com

FEMININE	MEAN SCORE	MASCULINE	MEAN SCORE
become nurse	-2.00	ask stupid question	0.20
make doll	-2.00	join circus	0.33
freshen up	-1.60	generate revenue	0.71
get assistance	-1.33	impress people	1.00
feel cold	-1.20	try solve problem	1.00
choose love	-0.80	enjoy power	1.29
pick berry	-0.60	see pretty girl	1.60
put weight	-0.40	catch football	2.00
resolve problem	0.00	want woman	2.00

Table 13: A stratified random sample of the gold standard.

total number of utterances for the action at hand. Formally, if we respectively denote the number of male and female occurrences of a particular verb phrase with m and f (subject to $m + f = n$) and the total number of male and female occurrences of all verb phrases with M and F (i.e., $M = \sum m$ and $F = \sum f$) then the final formula to compute the normalized score s of the given verb phrase becomes $s = (m - p)/\sigma_m$ where, $p = M/N$ and $\sigma_m = \sqrt{np(1-p)}$.

Note that, as a convention, I arbitrarily picked the sign of the measure so that a masculine (feminine) bias results in a positive (negative) score. A gender bias of 0 means the two genders are equally likely to mention a given verb phrase.

The rationale of using the proportions of gender utterances is that in Twitter, people talk about what they do; hence, if a certain verb phrase is used more often by one gender then I conclude that it is probably a more typical action of that gender.

The ukWaC corpus does not contain information about the gender of the people who use a given phrase. Instead, I employ a heuristic that uses the gender information of the pronouns and the proper names. Whenever I detect a verb phrase in a sentence, I look for the nearest pronoun or proper name (identified by the part of speech) in the sentence to the left side of the phrase and if it is a "he" or a male name ("she" or a female name) I count the occurrence of the phrase as a male (female) utterance – the gender of the names are guessed by using the same lists I employed for Twitter users. The gender bias is then computed by using the proportions of male and female utterances, in a similar fashion as for the Twitter corpus.

I should note that the two approaches are quite different in their nature. The Twitter approach directly taps the gender information about who uses a given verb phrase – regardless of who actually carries out the corresponding action. Whether people do (or like to do) what they tweet about frequently is an open empirical question. On the other hand, the ukWaC approach allows us to tap a larger amount of text and follows a more anecdotal path. It does not try to guess the gender of who utters a phrase, but the gender of the person that is said to perform the action described by the phrase. An initial analysis revealed that only half of the pronoun-associated verb-phrase utterances in Twitter are used with the first person singular pronoun "I" – suggesting that people not only talk about what they do but also about what other people do.

As a side note, one can think of other methods that make use of *stylometrics* and author-gender prediction techniques [Koppel et al., 2002] in order to guess the genders of authors of fragments of text observed in ukWaC. That would allow us to extract

mention-based gender scores of actions from Web-text similar to what we get from Twitter. I believe these are interesting avenues for future research.

5.4 COMMONSENSE COVERAGE

Although our focus is primarily on the actions contained in ConceptNet, it is informative to look at the coverage of the common sense (as it is represented in ConceptNet) in the corpora as a whole. The percentage of the commonsense concepts that are represented in ConceptNet and observed in corpora is quite high: Out of the original 267,364 ConceptNet phrases (including actions and non-actions), we observed 54% (145,486) in ukWaC and 51% (136,128) in ETC. If we focus on the actions then out of the 49,754 verb phrases, we observe 47% (23,455) in ukWaC, and 43% (21,442) in ETC.. Although the filtered ETC is much smaller than ukWaC, its coverage is about the same.

5.5 RESULTS

The top 10 masculine, feminine, and neutral actions computed over the two corpora are given in Table 14. The results in the table indeed look encouraging for the effectiveness of a corpus-based approach. Interestingly, there is no overlap between the two corpora – suggesting that Twitter and ukWaC may be covering different aspects of common sense. In the next three subsections, we will see a detailed evaluation of the gender bias. First, I report the correlation between the corpus-based predictions of gender score and the human gold standard. In the second subsection, I evaluate the predictive power of the gender bias on the direction of the stereotypical gender expectations of actions. After that, I carry out qualitative analyses that help us to interpret the data. In all cases, the Twitter and ukWaC scores are converted to z-scores.

5.5.1 *Spearman correlation*

In Table 15, I report the Spearman correlations between the gold standard and the gender biases computed by various methods. For each row, the number in parenthesis is the number of items covered by the corresponding method. *Combined* refers to taking the average of Twitter and ukWaC scores for each verb phrase. In this method, only the items that are covered both by Twitter and ukWaC are used. *Matching signs* reports the performance of combined method only on the verb phrases for which Twitter and ukWaC scores agreed on the sign of the gender biases.

	Masculine		Feminine		Neutral	
Rank	Twitter	ukWaC	Twitter	ukWaC	Twitter	ukWaC
1.	make money	do so	go bed	give birth	buy cheese	want much
2.	get free	take over	feel like	take place	chew food	ask yourself why
3.	want make money	think himself	want go	find out	go sea	build nuclear weapon
4.	make playoff	do in	feel good	become pregnant	wait area	build product
5.	unite state	do to	make smile	let know	take seat	carry everything
6.	earn money	keep himself	make cry	see herself	wait show	enter exit
7.	operate system	do what	go school	take part	lose track time	establish priority
8.	go down	become king	go sleep	provide information	make mind	prepare depart
9.	try out	go close	come home	think herself	become true	sell magazine
10.	come soon	raise up	see new	add basket	know appreciate	teach read write

Table 14: Top ranking gendered actions collected from Twitter and ukWaC. The neutral actions have the smallest absolute gender bias. Note that the two corpora do not have a single common word in the top ranking lists for either gender.

The correlation between the computed biases and gold standard is significant but quite low both for ukWaC and Twitter – 0.27 for the former and 0.28 for the latter. Taking the average of the Twitter and ukWaC-based biases seems to improve the correlation (0.33 for the combined case), but the highest correlation is observed when we focus on the verb phrases with matching signs (0.47). Of course, the increase in Spearman correlation comes with a cost: only on half of the verb phrases, the two methods agree on the polarity of gender.

METHOD	SPEARMAN	COVERAGE
ukWaC	0.27	98% (433)
Twitter	0.28	100% (441)
Combined	0.33	98% (433)
Matching signs	0.47	52% (231)

Table 15: Spearman correlations between various corpus-based scores and the gold standard. Coverage is the percentage of items that have an associated gender bias for the corresponding method.

5.5.2 *Predictive power of the gender biases*

To further assess the predictive power of the corpus-based scores, I transformed the gold standard into a two-class dataset by using the sign of the gender scores of the verb phrases as their labels – discarding the verb phrases with a human score of zero. Thus, *gold-standard* masculine and feminine phrases were labeled as "positive" and "negative", respectively. I used the sign of the gender biases based on Twitter and ukWaC as the predicted labels, and computed the area under the ROC curve (AUC) and accuracy of the predictions.

The results tabulated in Table 16 show us that it is possible to separate the feminine and masculine actions from each other with a reasonably high success. Especially if we limit ourselves to the phrases for which both Twitter and ukWaC scores agree on their signs, we can achieve an AUC of 0.76 and an accuracy of 0.70.

5.5.3 *Qualitative pattern analysis*

So far, we have seen that the corpus-based gender biases are not perfect predictors of the human gold standard but still, they are reasonably valuable indicators of stereotypical expectations. We

METHOD	AUC	ACCURACY	COVERAGE %
ukWaC	0.64	0.61	99% (375)
Twitter	0.65	0.61	100% (380)
Combined	0.67	0.59	99% (375)
Matching signs	0.76	0.70	54% (205)

Table 16: Classification performance of various corpus-based scores. Random baseline for both measures is 0.50. Coverage is the percentage of items that have an associated gender bias for the corresponding method.

were able to obtain a Spearman correlation of 0.46 and an AUC of 0.76 on a restricted subset of the dataset (more than half of the ConceptNet sample). Considering that we have more than 20,000 verb phrases in the general dataset sampled from ConceptNet, we can expect to gender-tag approximately 10,000 actions with a certain reliability.

Another important point is that the errors are instructive. In Figure 7, we see the scatter plot of Twitter versus ukWaC scores of the verb phrases and we observe that the two methods have quite different results. The Spearman correlation between the scoring methods is only 0.19.

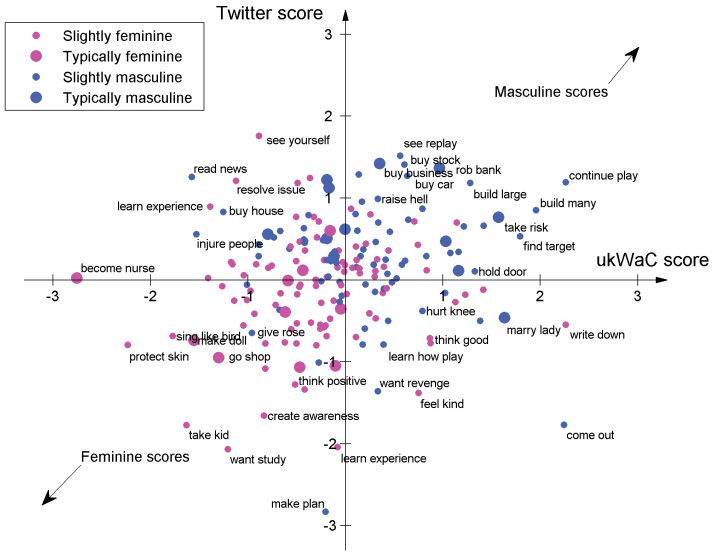

Figure 7: Scatter plot of ukWaC bias (x-axis) versus Twitter bias (y-axis); for both axis, higher positive values represent higher masculine gender bias. Color coding represents the polarity of the human gold standard and size is proportional to the magnitude of average human score.

The first and third quadrants of the figure, where the signs of Twitter and ukWaC biases match, contain a majority of gold-standard masculine and feminine actions, respectively. This is not surprising as we already saw the increased performance of the matching-signs condition. It is more interesting to look at the mismatch between Twitter and ukWaC. The actions that are placed in the second and fourth quadrants have mismatching signs of Twitter and ukWaC biases. A plausible interpretation for such items is that they are the actions that one gender talks about much, but is reported to be doing it less often. Consider "resolve issue" in the fourth quadrant for instance – an action that is rated as slightly feminine by the human raters: in ukWaC, females are reported to be "resolving issues" more often than males (hence a negative ukWaC bias), whereas in Twitter, males mention this action more frequently (hence a positive Twitter bias). A similar case is "come out" which is placed in the second quadrant – an action that is rated as slightly masculine by the human raters: in ukWaC, males are reported to be "coming out" more often, whereas it is female Twitter users who mention this action more frequently.

Another informative visualization is given in Figure 8 which is a scatter plot of the human gold standard versus Twitter bias. In this figure, I plot only the actions with a sign mismatch. We can interpret these actions as the actions that one gender talks about much (mentions it in Twitter more frequently than the other gender does) but are rated to be associated with the other gender, by the human raters. For example, the actions *take note* and *get assistance* are mentioned more often by males in Twitter but they are considered to be feminine by the human raters. The phrases *build snowman* and *want revenge* are examples of the gold-standard masculine actions that females talk about.

In sum, the qualitative analysis suggests a more complex picture, in which mismatches (between Twitter and ukWaC, as well as corpora and the human gold standard) are not necessarily mistakes of the text-based methods, but a sign that we are tapping into different kinds of information: explicit assessments of stereotypical actions (gold standard), actions that males and females are reported as being doing in natural written discourse (ukWaC), and actions that they like to talk about (Twitter). Each of these sources might be useful for different purposes.

5.6 CONCLUSION

We have seen novel ways that utilize the metadata contained in a Twitter corpus and simple linguistic cues in the ukWaC Web corpus in order to extract stereotypical expectations about

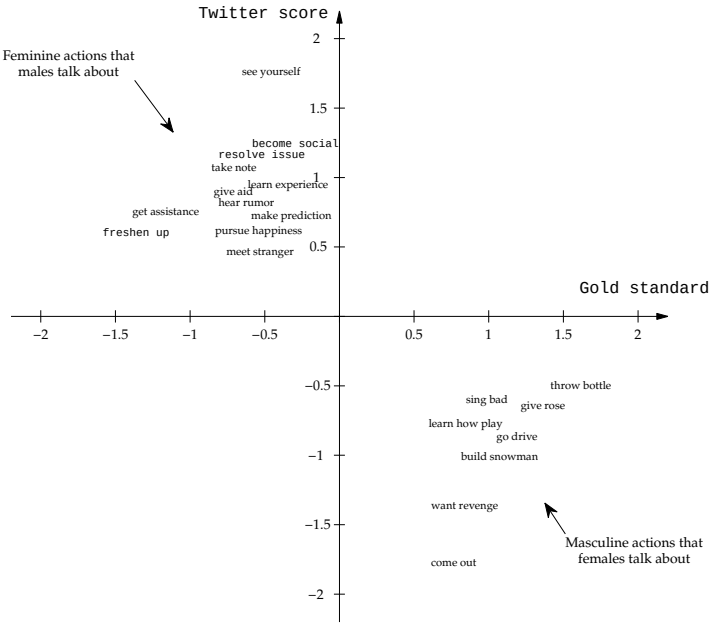

Figure 8: Scatter plot of Twitter bias (y-axis) versus the gold standard (x-axis). Only the actions that are in the second and fourth quadrants are shown.

actions that are pertinent to common sense. While working on this problem, we observed that both ukWaC and Twitter have a wide coverage of commonsense concepts. More than half of the unique concepts in ConceptNet were detected in the two corpora. The gender-filtered Twitter corpus is much smaller than ukWaC, but has an equally wide coverage of common sense.

I conclude that both the metadata about the Twitter users and corpus-based gender attribution heuristics definitely help in stereotypical knowledge mining. On several performance metrics, the Twitter-based approach is at least as good as the ukWaC-based scoring system. Moreover, combining the two methods works even better.

Apart from predicting the gender expectation of a given verb phrase (which seems feasible, based on the experiments I reported), the methodology may allow us to dig in deeper and provide refined data that might be used in sociolinguistics, gender studies, and personalized information retrieval and recommendations. The demographic dimensions we can extract from Twitter (or other similar social media) are not limited to gender. Time of the day, geographical location, and other metadata can be employed in similar ways to augment commonsense knowledge repositories, and equip computers with an even better understanding of how humans work.

6 CONCLUSION

J.-B. Clamence: We are not certain, we are never certain. If we were we could reach some conclusions, and we could, at last, make others take us seriously.

— Albert Camus, The Fall

The center theme of this thesis has been collecting commonsense from two very different sources, people and textual data. While pursuing different methods to achieve this aim, I proposed and evaluated 1) a text mining algorithm, BagPack, 2) a game with a purpose, Concept Game, and 3) novel methods to extract stereotypical common sense from Web corpora.

Below, I repeat the four research questions that I set at the beginning of the discourse and discuss what we have learned while looking for their answers.

RESEARCH QUESTION 1 Do corpora, and Web-based corpora in particular, contain commonsense knowledge, and if so can we extract it?

In Section 3.5, we saw that it is possible to mine commonsense assertions from Web-based corpora and obtain a performance that is significantly better than random baseline by using machine learning models – that are trained, also, on Web-based corpora. Although the performance is far from perfect, it is comparable to the state of the art in several commonly used semantic tasks.

Moreover, in Section 5.4, we observed that more than half of the commonsense concept phrases that are represented in OMCS are also observed in the Web-based corpus (ukWaC) and the Twitter-based corpus (ETC). I should note that ETC is a smaller corpus compared to ukWaC but its commonsense coverage is almost at the same level with that of ukWaC.

Thus, I conclude that the answer for the first research question is yes and yes: Web-based corpora contains commonsense knowledge and, more importantly, it is possible to extract this knowledge by means of appropriate methods including text mining. The extent of the common sense contained in corpora is yet to be determined with future studies.

RESEARCH QUESTION 2 Can we combine text mining and human computation in order to achieve a better commonsense collection?

and its corollary:

RESEARCH QUESTION 3 Is it possible to collect high-quality commonsense assertions from people while they play a fast-paced game where the players' main motivation is having fun?

I carried out two experiments to address these questions. In the first one, presented in Section4.3, we saw that the output of text miner can be greatly improved – in terms of AUC – by passing the candidates through Concept Game and having players rate the assertions. Collecting three responses per assertion from the players and employing a majority rule seems enough to achieve an accuracy which is equivalent to that of OMCS, that was created by volunteers.

In Section 4.4, we saw another experiment where the output of Concept Game is used as the training set for text miner. This setting allows us to assess the quality of the output of Concept Game in a relatively "external" task where we not only assess the quality of output directly but also its effectiveness as a training set indirectly. The results suggest that bootstrapping leads to a performance in text mining that is at least as good as the one obtained with original ConceptNet-based training set.

Before I conclude on this question, I should note that another important advantage of combining text mining and human computation is that the domain of new assertions is not limited to what people would provide in a pure human-computational approach – where it is the people that produce the assertions – but we can tap the large amount of texts to collect assertions about very diverse topics and then have them filtered by people. Presumably, that allows us to collect assertions about concepts that humans do not tend to state explicitly when they are asked to. This advantage was reflected in the fact that 23% of the concepts that are mined were not attested in the entire OMCS knowledge base.

RESEARCH QUESTION 4 Can we extract subjective aspects of common sense like the stereotypical expectations or prejudices of people from Web corpora?

The question of whether we should include stereotypical knowledge in machine-readable knowledge bases of common sense is interesting, but regardless of our answer to this question, we need to be able to identify such knowledge – either to explicitly keep it in the knowledge base or avoid keeping it implicitly. My focus in Chapter 5 has been on extracting gender-related stereotypical knowledge from corpus as a case study.

We saw how corpus-based techniques can be employed in extracting gender-related stereotypical knowledge from text. My proposal utilizes large amount of text collected from Web (ukWaC) and social media (in the specific case, Twitter), and combines it

with linguistic heuristics and meta data about the users to predict stereotypical expectations about actions. More specifically, I employed heuristics that are based on the first names of Twitter users and gendered names and pronouns used in Web-based text. Such techniques allow us to capture which gender mentions a particular option more often and which gender is reported to be carrying out the same action separately.

For evaluation, I created a sample of actions represented in OMCS and predicted which actions are perceived as masculine and feminine by human raters. The results are very encouraging: There is a substantial correlation between the human scores and the predicted scores and the corpus-based predictions obtained a high accuracy in predicting the polarity of human raters's scores. Moreover, the combination of Twitter and Web-based predictions provide a higher performance.

FINAL REMARKS The task of creating a commonsense knowledge base is an ambitious one and one has to deal with many facets of the commonsense collection problem. Crowd-sourcing techniques allow us to tap laypeople's "expertise" on common sense, but we must be aware of the fact that the way we present the task to people has a direct effect on the type and quality of the data we will collect. In this thesis, I addressed one particular area of the design space of games with a purpose: Concept Game does not rely on people for the production of common sense, but for the verification of it. And while doing so, it works in tandem with text mining methods that parse large amount of texts where humans produce commonsense implicitly while performing natural communicative tasks. The text mining technique that I proposed in this thesis, BagPack, is one such a technique which represents the relations between concepts in a vector space.

As I said, there are many facets to the commonsense collection issue, and the problem of subjective knowledge is one of them. Common sense does not only consist of objective statements about the world we live in but also contains our prejudices and represents our stereotypical expectations about various conditions. I wanted to draw attention to this aspect of the problem, and as a case study I showed how we can put corpus-based techniques into service to the task of collecting stereotypical gender expectations of actions.

The commonsense problem in its widest sense has been and will be a central topic of investigation in the foreseeable future because any step that we can take towards the solution of it will have important ramifications in all areas of AI – including but not limited to natural language understanding, human computer interaction, information retrieval and planning. Moreover, the fact

that engineering a system that can represent, collect, and process commonsense knowledge amounts to engineering a human-level AI system tells us that there are many more steps that we can (must) take before proclaiming the problem solved. I dare say the results and the experiences that I reported in this thesis will prove useful to those who wish to pursue this problem in the future.

APPENDIX

A

RESOURCES

A.1 PLAYER RESPONSES OF CONCEPT GAME

The output of Concept Game is released into the public domain. The responses of players for the candidate assertions can be found at http://github.com/amacinho/Concept-Game-Datasets. I plan to continue releasing the output as more responses are collected.

A.2 GOLD STANDARD OF STEREOTYPES

The gold standard for the gender expectations of actions are collected by using Crowdflower's crowd-sourcing services as explained in Section 5.2.3. The gold standard dataset is available at http://github.com/amacinho/Gender-Expectations. For illustrative purposes, top ranking 20 feminine and 20 masculine actions are given in Table 17.

A.3 GENDER GUESSING VIA NAMES

The lists of most popular American names come from two sources US Census Bureau[1] and US Social Security Administration([2]. While the lists can be constructed by using the public data from these sources, I combined them into a single dataset and released a set of Python scripts that makes it easier to guess the gender of given name. The dataset and the scripts can be downloaded at http://github.com/amacinho/Name-Gender-Guesser.

A.4 CORPUS-BASED GENDER SCORES OF ACTIONS

In this thesis, for evaluation purposes, I focused on a small sample of the actions that are represented in OMCS. However, the corpus-based methods discussed in Section 5.3 compute the gender biases of all of the actions that are represented in OMCS. In Tables 18 and 19, I tabulate the top 20 masculine and feminine actions according to the Twitter and ukWaC-based scores. The dataset containing all 21,442 actions extracted from OMCS can be downloaded at https://github.com/amacinho/Gender-Expectations-Predictions/

1 http://www.census.gov/
2 http://www.ssa.gov/

RANK	FEMININE	MASCULINE
1.	become nurse	catch football
2.	make doll	marry lady
3.	accept proposal	find woman
4.	remember date	buy business
5.	go shop	take risk
6.	tell how feel	see pretty girl
7.	make beautiful	rob bank
8.	freshen up	kill other
9.	make nostalgic	kill bird
10.	want shop	catch mouse
11.	fill house	throw puncho
12.	hold hair back	steal car
13.	make sauce	put jail
14.	feel warmth	join army
15.	make people happy	throw bottle
16.	spend time people	sell sell
17.	protect skin	hold door
18.	get assistance	build many
19.	make up	compete against other
20.	feel cold	add oil

Table 17: A list of the most feminine and most masculine actions according to the gold standard.

Masculine		
Rank	Twitter	ukWaC
1.	make money	do so
2.	get free	take over
3.	want make	money do in
4.	make playoff	do to
5.	unite state	think himself
6.	earn money	do what
7.	operate system	come up
8.	go down	keep himself
9.	try out	get the
10.	come soon	write book
11.	learn how	take up
12.	achieve goal	do the
13.	find out truth	pick up
14.	sell product	go close
15.	install window	become king
16.	create job	make clear
17.	write code	raise up
18.	discover answer	do it
19.	keep up good work	join army
20.	follow up	do and

Table 18: Top ranking masculine actions based on Twitter and ukWaC.

	Feminine	
Rank	Twitter	ukWaC
1.	go bed	give birth
2.	feel like	take place
3.	want go	become pregnant
4.	feel good	find out
5.	make smile	let know
6.	make cry	see herself
7.	go school	think herself
8.	go sleep	provide information
9.	come home	add basket
10.	see new	take part
11.	make feel	develop country
12.	make laugh	incorporate business
13.	go out	provide support
14.	go home	do your
15.	make happy	raise fund
16.	go see	bring together
17.	go away	leave husband
18.	go back	raise money
19.	harry potter	become mother
20.	come true	make available

Table 19: Top ranking feminine actions based on Twitter and ukWaC.

BIBLIOGRAPHY

S. Argamon, M. Koppel, J. Fine, and A. Shimoni. Gender, Genre, and Writing Style in Formal Written Texts. *Text*, 23:3, 2003.

S. Argamon, M. Koppel, J. Pennebaker, and J. Schler. Mining the blogosphere: age, gender, and the varieties of self-expression. *First Monday*, 12(9), 2007.

M. Banko, M. Cafarella, S. Soderland, M. Broadhead, and O. Etzioni. Open information extraction from the Web. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 2670–2676, Hyderabad, India, 2007.

M. Baroni and A. Lenci. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 2010. In press.

M. Baroni, S. Bernardini, A. Ferraresi, and E. Zanchetta. The WaCky wide web: A collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226, 2009.

E. Biçici and D. Yuret. Clustering word pairs to answer analogy questions. In *Proceedings of the Fifteenth Turkish Symposium on Artificial Intelligence and Neural Networks*, pages 277–284, Muğla, Turkey, 2006.

T. Brants and A. Franz. Web 1T 5-gram Version I, 2006.

P. Buitelaar and P. Cimiano. *Bridging the Gap between Text and Knowledge*. IOS, Amsterdam, 2008.

S. Canu, Y. Grandvalet, V. Guigue, and A. Rakotomamonjy. Svm and kernel methods matlab toolbox, 2005.

N. Chomsky, editor. *Syntactic structures*. Mouton, The Hague, 1957.

K. Church and P. Hanks. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1): 22–29, 1990.

K. Erk. A simple, similarity-based model for selectional preferences. In *Proceedings of ACL*, pages 216–223, Prague, Czech Republic, 2007.

I. Eslick. *Searching for Commonsense*. Ms thesis, MIT, Cambridge, MA, 2006.

T. Fawcett. An introduction to ROC analysis. *Pattern Recogn. Lett.*, 27(8):861–874, 2006. ISSN 0167-8655. doi: http://dx.doi.org/10.1016/j.patrec.2005.10.010.

J. Gordon, B. V. Durme, and L. Schubert. Evaluation of commonsense knowledge with mechanical turk. In *In NAACL Workshop on Creating Speech and Language Data With AmazonŠs Mechanical Turk*, 2010a.

J. Gordon, B. Van Durme, and L. Schubert. Learning from the Web: Extracting general world knowledge from noisy text. In *Proceedings of the AAAI 2010 Workshop on Collaboratively-built Knowledge Sources and Artificial Intelligence*. ACM, 2010b.

C. Havasi, R. Speer, and J. Alonso. ConceptNet 3: a flexible, multilingual semantic network for common sense knowledge. In *Recent Advances in Natural Language Processing*, 2007.

C. Havasi, R. Speer, J. Pustejovsky, and H. Lieberman. Digital intuition: Applying common sense using dimensionality reduction. *IEEE Intelligent Systems*, 24(4):24–35, 2009.

J. Holmes. Hedges and boosters in women's and men's speech. *Language & Communication*, 10(3):185–205, 1990.

M. Koppel, S. Argamon, and A. Shimoni. Automatically categorizing written texts by author gender. *Literary and Linguistic Computing*, 17(4):401–412, 2002.

R. Lakoff. Language and woman's place. *Language in society*, 2 (01):45–80, 1973.

D. Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 11:33–38, 1995.

D. B. Lenat. From 2001 to 2001: Common sense and the mind of HAL. In D. Stork, editor, *HAL's Legacy: 2001's Computer as Dream and Reality*, chapter 9, pages 193–209. MIT Press, 1996.

H. Lieberman. Usable AI requires commonsense knowledge. In *Workshop on Usable Artificial Intelligence, ACM Conference on Computers and Human Interaction (CHI-08), Florence, Italy*, 2008.

H. Lieberman, D. Smith, and A. Teeters. Common consensus: a web-based game for collecting commonsense goals. In *Proceedings of IUI07*, Honolulu, HI, 2007.

H. Liu and R. Mihalcea. Of men, women, and computers: Data-driven gender modeling for improved user interfaces. In *Proceedings of the International Conference on Weblogs and Social Media (ICWSM)*, Boulder, Colorado, 2007.

H. Liu and P. Singh. Commonsense reasoning in and over natural language. In *Knowledge-Based Intelligent Information and Engineering Systems*, pages 293–306. Springer, 2004.

C. Matuszek, M. Witbrock, R. Kahlert, J. Cabral, D. Schneider, P. Shah, and D. Lenat. Searching for common sense: Populating Cyc from the Web. In *Proceedings of the Twentieth National Conference on Artificial Intelligence*, volume 3, pages 1430–1435. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

J. McCarthy. Programs with common sense. In *Proceedings of Teddington Conference on the Mechanization of Thought Processes*, pages 75–91, London, England, 1959.

M. Minsky. Commonsense-based interfaces. *Communications of the ACM*, 43(8):66–73, 2000.

U. Padó. *The Integration of Syntax and Semantic Plausibility in a Wide-Coverage Model of Sentence Processing*. Dissertation, Saarland University, Saarbrücken, 2007.

U. Padó, S. Padó, and K. Erk. Flexible, corpus-based modelling of human plausibility judgements. In *Proceedings of EMNLP*, pages 400–409, Prague, Czech Republic, 2007.

P. Pantel and M. Pennacchiotti. Espresso: Leveraging generic patterns for automatically harvesting semantic relations. In *Proceedings of COLING-ACL*, pages 113–120, Sydney, Australia, 2006.

S. Petrović, M. Osborne, and V. Lavrenko. The Edinburgh Twitter corpus. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Linguistics in a World of Social Media*, pages 25–26. Association for Computational Linguistics, 2010.

J. Quesada, P. Mangalath, and W. Kintsch. Analogy-making as predication using relational information and LSA vectors. In *Proceedings of CogSci*, page 1623, Chicago, IL, USA, 2004.

A. J. Quinn and B. B. Bederson. A taxonomy of distributed human computation. Technical Report HCIL-2009-23, University of Maryland, College Park, 2009.

W. Rafelsberger and A. Scharl. Games with a purpose for social networking platforms. In *HT '09: Proceedings of the 20th ACM conference on Hypertext and hypermedia*, pages 193–198, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-486-7. doi: http://doi.acm.org/10.1145/1557914.1557948.

D. Ramachandran, P. Reagan, and K. Goolsbey. First-orderized researchcyc: Expressivity and efficiency in a common-sense ontology. In *AAAI Workshop on Contexts and Ontologies: Theory, Practice and Applications*, 2005.

M. Sahlgren. *The Word-Space Model*. Dissertation, Stockholm University, Stockholm, 2006.

L. Schubert and M. Tong. Extracting and evaluating general world knowledge from the Brown corpus. In *Proceedings of the HLT-NAACL 2003 workshop on Text meaning-Volume 9*, pages 7–13. Association for Computational Linguistics Morristown, NJ, USA, 2003.

C. Sherron. Constructing common sense. In E. Balka and R. Smith, editors, *Women, work and computerization: Charting a course to the future*, pages 111–118. Kluwer Academic Publishers, 2000.

B. Smith. Formal ontology, common sense and cognitive science. *International Journal of Human Computer Studies*, 43(5):641–668, 1995.

R. Speer. Open Mind Commons: An inquisitive approach to learning common sense. In *Proceedings of the Workshop on Common Sense and Intelligent User Interfaces*, Honolulu, HI, 2007.

R. Speer, C. Havasi, and H. Surana. Using verbosity: Common sense data from games with a purpose. In *Proceedings of the Twenty-Third International Florida Artificial Intelligence Research Society Conference (FLAIRS 2010)*, pages 104–109, 2010.

M. Strohmaier and M. Kröll. Studying databases of intentions: do search query logs capture knowledge about common human goals? In *Proceedings of the fifth international conference on Knowledge capture*, pages 89–96. ACM, 2009.

P. Turney. Similarity of semantic relations. *Computational Linguistics*, 32(3):379–416, 2006a.

P. Turney. Expressing implicit semantic relations without supervision. In *Proceedings of COLING-ACL*, pages 313–320, Sydney, Australia, 2006b.

P. Turney. A uniform approach to analogies, synonyms, antonyms and associations. In *Proceedings of COLING*, pages 905–912, Manchester, UK, 2008.

P. Turney and M. Littman. Corpus-based learning of analogies and semantic relations. *Machine Learning*, 60(1-3):251–278, 2005.

P. Turney, M. Littman, J. Bigham, and V. Shnayder. Combining independent modules in lexical multiple-choice problems. *Recent Advances in Natural Language Processing III: Selected Papers from RANLP*, 2003:101–110, 2003.

L. Vanderwende. Volunteers created the web. In *2005 AAAI Spring Symposium on Knowledge Collection from Volunteer Cotnributors (KCVC'05)*, 2005.

L. Von Ahn. Games with a purpose. *Computer*, 29(6):92–94, 2006.

L. Von Ahn, M. Kedia, and M. Blum. Verbosity: A game for collecting common-sense knowledge. In *Proceedings of CHI*, pages 75–78, Montreal, Canada, 2006.

Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 42–49. ACM, 1999. ISBN 1581130961.

C.-H. Yu and H.-H. Chen. Commonsense knowledge mining from the web. In *Proceedings of AAAI 2010*, 2010.

DECLARATION

I hereby declare that this thesis is a presentation of my original research work. Wherever contributions of others are involved, I made every effort to acknowledge them with due reference.

The work was done under the supervision of Professor Marco Baroni, at the Doctoral School in Cognitive and Brain Sciences of Trento University, Italy.

Rovereto, November 2010

Amaç Herdağdelen