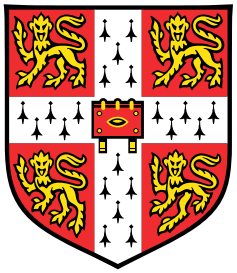
Improving Automated Literature-based Discovery with Neural Networks

Neural biomedical Named Entity Recognition, Link Prediction and Discovery



Gamal Kashaka Omari Crichton

Theoretical and Applied Linguistics
University of Cambridge

This dissertation is submitted for the degree of
Doctor of Philosophy

St. Edmund's College

February 2019

To my mother, Ursula Juliana Crichton, for love, encouragement and the countless years and ways she prepared me to pursue this degree and live life well.

To my brother, Maceo Amadi Crichton, for being a source of support and encouragement throughout this degree and life in general.

To other family members who offered support and encouragement, especially Roamel, Gideon, Uncle Noel (late), Aunties Venus and Lynette, Uncle Benjamin and cousins.

Declaration

I hereby declare that this dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except as specified in the text. It is not substantially the same as any that I have submitted, or, is being concurrently submitted for a degree or diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. I further state that no substantial part of my dissertation has already been submitted, or, is being concurrently submitted for any such degree, diploma or other qualification at the University of Cambridge or any other University or similar institution except as declared in the Preface and specified in the text. This dissertation contains fewer than 80,000 words including footnotes, references and appendices, but excluding bibliography.

Gamal Kashaka Omari Crichton
February 2019

Acknowledgements

I thank God for being my sustenance, guide and everything else he promised in his word to be to those who love him, throughout this degree and for the ways he has grown me in all aspects through this experience.

I thank Prof. Anna Korhonen for being a compassionate and competent Supervisor who cared about my well-being and development and for being instrumental in securing funding throughout my time here. I also thank my co-supervisor Dr. Nigel Collier and the other members of the PhD monitoring panels for providing helpful feedback and general research direction.

I thank my Christian brothers and sisters in the Christian Graduate Society (CGS), especially Tuesday Bible Study, and at Eden Baptist Church. Particular CGSers include Ben, Feri, Clara, John and Di, Johan, Danny, Ken, Kimberly, Paula, Mariëtta, Marcos and Dana, Solène, Nard, Alex, Godwin, Rita, Bruno, Jane, Derk; along with all those who co-served on the CGS Committee and the International Student Welcome (ISW) Committees with me. You have been a blessing and a continual source of spiritual encouragement.

I thank the members of the 11 Roseford Road community who I was lucky to share it with: Ulrich, Theresia, Jess, Patrick, Talia, Mark, Blaise and Ronja. You taught me how living in a Christian community works, showed the fruits of such living and helped me to mature as a person.

I thank the various Post Doctoral members of the Language Technology Laboratory (LTL) who aided my development as a researcher. Special thanks to Sampo for much training and direction; Yufan for ideas and assistance; Simon for discussions and assistance; Taher and Ivan for tips on successful research mindset and practices; Ehsan for reading and refining parts of the thesis; and Marek for collaboration and help.

I thank my PhD compatriots during my time at LTL: Daniela, Billy, Milan, Edoardo, Yi, Victor, Olga, Costanza, Flora and Yan for help and research discussions and for being friends, encouragers and all round great lab-mates. Thanks also to the other (D)TAL PhDs/post-docs/staff who helped make my time here memorable.

I thank pre-Cambridge friends for encouragement and continued relationships, especially Attica, Te-jé, Terry, Xavier and others who sent greetings.

Abstract

Literature-based Discovery (LBD) uses information from explicit statements in literature to generate new or unstated knowledge. Automated LBD can thus facilitate hypothesis testing and generation from large collections of publications to support and accelerate scientific research, which is adversely affected by publication explosion and knowledge fragmentation. Existing methods, however, use methodologies which are inadequate for capturing the complex information available in scientific literature and are prone to proposing spurious discoveries or an abundance of low-quality ones. To be capable of solving these problems, automated LBD needs to accurately glean the extensive information present in literature, cope with the dynamic nature of scientific knowledge and place high-quality proposals at the top of ranked outputs.

Recent advances in Natural Language Processing (NLP) allow for deep textual analysis to obtain a wide coverage of information present in text and can adapt easily to recognising new biomedical entities and terms. Similarly, recent advances in graph processing have made it possible to do in-depth analysis on information represented as graphs, such as published biomedical connections, to facilitate high-quality knowledge discovery. Both of these advances utilise neural networks extensively.

This work used neural networks in a bid to advance automated LBD in three ways: 1) improving biomedical Named Entity Recognition (NER) to extract entities from unstructured text by using multi-task learning across multiple biomedical datasets; 2) improving knowledge discovery from realistic, random- and time-sliced biomedical graphs using link prediction and 3) improving the ranking of published discoveries on open- and closed- LBD instances by scoring the strength of connection paths using neural models. Excitingly, the latter approaches outperformed those used by the state-of-the-art LION LBD system, indicating that their integration into it would provide better support to cancer researchers using it.

The results from this work show that it is feasible to use neural networks to improve LBD in different ways. They also demonstrate that neural networks are versatile enough to be applied to improve traditional as well as non-traditional LBD. The principal implication of these findings is that neural biomedical knowledge discovery, especially LBD, is presently useful in addition to being a potentially rich field for further study.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Automated Literature-based Discovery (LBD) and its Potential

Literature-based Discovery (LBD) aims to discover new knowledge by connecting information which are explicitly stated in literature to deduce connections which are not explicitly stated in literature. The field was pioneered by Don Swanson who hypothesised that the combination of two separately published results indicating that "A causes B" and "B causes C" are evidence of a relationship between A and C which is usually unknown or unexplored. He used this method to propose fish oil as a treatment for Raynaud syndrome based on their shared connections to blood viscosity in published literature (Swanson, 1986a). This hypothesis was later shown to have some merit in a prospective study (DiGiacomo et al., 1989) and along with Neil Smalheiser, he continually proposed other discoveries using similar methods including between migraine and magnesium (Swanson, 1988), Somatomedin C and arginine (Swanson, 1990b), Alzheimer's disease and Estrogen (Smalheiser and Swanson, 1996b), and several others (Smalheiser and Swanson, 1996a, 1998).

Since LBD generates new knowledge by combining existing literature and has shown potential using a mostly manual approach, the possibility of using computers and algorithms to discover many such connections automatically in large collections of literature is tantalising. This is called *automated LBD* and it can facilitate both complex hypothesis testing and hypothesis generation from large collections of literature and thus give tangible support to scientific research (Hristovski et al., 2013; McDonald et al., 2012). Scientific literature is growing at an exponential rate (Hunter and Cohen, 2006) making it difficult for researchers to stay current in their discipline. This overwhelming volume of publications and the increasing necessity of researchers to specialise has led to non-interacting literature silos, which creates

an environment where discoveries in one area are not known outside of it (Swanson, 1990a) and valuable logical connections between disparate bodies of knowledge remain unnoticed (Swanson, 1986b). This creates a situation where there is a very real chance that pieces of information which can be combined to make breakthroughs are already discovered but are splintered and dispersed in the literature. Automated LBD can solve these problems by helping researchers to quickly gain information on relevant advances inside and outside of their respective niches and increase interdisciplinary information sharing. Thus, as the scientific literature grows, automated LBD is becoming an increasingly necessary research tool. For the rest of this thesis we will deal with only automated LBD, which we will refer to simply as LBD.

LBD has already proven its usefulness in a range of applications. It has been used to identify new connections between biomedical entities (such as genes, drugs and diseases); new candidate genes and treatments for illnesses (Hristovski et al., 2013); and to propose treatments for Parkinson's Disease, Multiple Sclerosis and cataracts (Kostoff, 2008b; Kostoff and Briggs, 2008; Kostoff et al., 2008a). It has seen use in drug development and repurposing (Ahlers et al., 2007; Hristovski et al., 2010; Zhang et al., 2014), as well as predicting adverse drug reactions (Banerjee et al., 2014; Shang et al., 2014). It has also been used to propose new potential cancer treatments (Ahlers et al., 2007). Its use has also been explored outside the biomedical domain, where it has been applied to developing water purification systems, accelerating development of developing countries and identifying promising research collaborations (Gordon and Awad, 2008; Hristovski et al., 2015; Kostoff et al., 2008c).

Despite these promising applications and its potential for knowledge discovery and increased research efficiency, at present LBD systems are yet to see widespread adoption and any meaningful uptake by those who can potentially benefit the most from them (Henry and McInnes, 2017; Kostoff, 2008a). There are several reasons for this which include non-technical issues like lack of interaction between developers and users during development which leads to systems which are incompatible with the workflow of researchers, but there are also technical shortcomings in existing LBD approaches which negatively impact their performance and hinder their application in real-world environments.

1.2 LBD Shortcomings and Possible Solutions

LBD is a secondary process because it needs to utilise the outputs of other processes to discover new knowledge. Broadly speaking LBD must have concepts (entities) and some notion of which concepts are related and which are not. There are multiple ways that

concepts can be procured and an almost infinite amount of relationships can be defined which determine how they are related. There are shortcomings at the stage of producing concepts and relationships as well as at the stage of using them to actually propose discoveries.

1.2.1 Concept and Relationship Procurement Stage

Current methods to procure concepts are based on methodologies such as matching text with entries in ontologies or dictionaries. Among other things, these methodologies inherently limits the captured information to the content of these resources instead of the larger amount of information available in literature. Such static methodologies are also prone to becoming outdated and require repeated expensive human and capital investment to maintain and update. Similarly, for relationships between concepts, existing methods rely on indications like literature co-occurrence which can produce spurious and noisy connections while missing genuine connections expressed as synonyms, among other infractions (Hristovski et al., 2006; Preiss et al., 2012). These can lead to substandard input to the discovery phase which can have a detrimental effect on the performance of LBD approaches.

Natural Language Processing (NLP) combined with Text Mining (TM) allow for deeper text analysis and could present opportunities for much wider coverage of concepts and relationships present in published literature. There has been widespread application of TM and NLP to biomedicine which has produced tools for tasks such as literature curation and semantic database development (McDonald et al., 2012; Simpson and Demner-Fushman, 2012). It thus seems likely that they could support LBD (Hristovski et al., 2013; McDonald et al., 2012), but there is presently little work on this (Preiss et al., 2012; Tsuruoka et al., 2011).

The better inputs produced by NLP and TM can lead to more dynamic LBD systems which are better capable of evolving with science, so research on improving LBD have focused on both the concepts and relations for use in LBD. There are indications that this is the future of the field and there are works which use NLP methodologies to perform these tasks, but especially so in procuring and processing concepts. Procuring and processing concepts with NLP involves *Named Entity Recognition (NER)* to obtain concepts from weakly structured text along with normalization and entity linking to ground and disambiguate them.

1.2.2 Discovery Stage

Existing LBD methods usually suffer from one or more shortcomings in the discovery phase which inhibit their capabilities. The most prominent ones include failing to demonstrate that they are capable of scaling to perform discovery on a large scale; producing highly

ranked useful discoveries; and performing well beyond the limited traditional paradigm which dominates the field. These shortcomings, respectively, lead to systems only being feasible for performing discoveries on a small amount of entities; proposing an overwhelming amount of low-quality discoveries; and only proposing discoveries which are so logically simple that they are easily deduced by humans or which are already common knowledge. We propose that *link prediction* can help overcome these and other problems in the discovery phase of LBD.

Link prediction involves predicting links or edges between nodes in a graph which are currently not present in it. If there is a graph whose nodes represent concepts and whose edges represent literatue-induced relationships between the concepts, then this graph effectively represents the current state of the literature. Link prediction on such a graph is analogous to LBD. There have been works using link prediction as an approach to LBD (Eronen et al., 2012; Katukuri et al., 2012; Sebastian et al., 2015).

However, link prediction is also independently a valid form of knowledge discovery and one which can be more powerful than LBD as it is freed from the assumptions of explicit connectivity which define LBD. That is, for a given graph representing knowledge contained in literature, LBD is link prediction on a subset of nodes which meet the additional criterion that a path consisting of no more than a stipulated maximum number of edges exists. This makes link prediction as an avenue for biomedical knowledge discovery a particularly fertile area for current research as it can be constrained to perform traditional LBD and unleashed to perform general biomedical knowledge discovery. The case for this is strengthened by the plethora of recent works which have focused on using neural networks and deep learning to produce improved graph representations and perform graph-related tasks such as link prediction and node classification. In this work, both of these uses of link prediction are explored with promising results for their application to LBD and general biomedical knowledge discovery.

This thesis presents research to improve the performance in both stages of LBD. Improving biomedical NER is used to advance the first stage while link prediction on biomedical graphs is its counterpart in the discovery phase. Machine learning models, some inspired by work on the latter, are also applied to evaluations used in a real-world, recently-released LBD system which uses a traditional approach to LBD. The work presented in this thesis relies on recent advances in neural networks in several areas including improved development and training of deep neural models and enhanced word and graph representations.

1.3 Neural Networks, Deep Learning and LBD

Machine Learning (ML) is concerned with finding patterns in large amounts of data to successfully complete a given task. Traditional ML techniques were limited in their ability to process data in their raw form and so constructing ML systems required in-depth domain expertise to design and engineer feature extractors. The feature extractors would discover the relevant features of the input data to create feature vectors which were then used by the machine learning system to detect or classify patterns in the input. Neural networks are universal approximators co-opted for representation learning which allow machines to take the raw data as input directly and automatically produce the representations needed for detection or classification. Deep learning methods are neural representation learners with multiple levels obtained by stacking non-linear modules. By composing enough of these modules, more complex functions between the input and the output can be learned (LeCun et al., 2015).

Since ML aims to find patterns in large amounts of data and LBD seeks to uncover discovery patterns from large amounts of literature, at a high level there is a case for the use of neural networks in LBD. A unified deep learning system for LBD may be something for the future, but as a stepping stone to that, at present these tools and methods can be used to improve various parts of the current LBD machinery. They have been applied to concept and relation extraction from text (Chiu and Nichols, 2016; Lample et al., 2016; Rei et al., 2016), to create high-quality representations of graphs (Grover and Leskovec, 2016; Ou et al., 2016; Perozzi et al., 2014; Tang et al., 2015; Wang et al., 2016) and to link prediction (Grover and Leskovec, 2016; Wang et al., 2016). In light of the fact that they have been applied to these LBD-related tasks and report state-of-the-art performance, it is conceivable that they could be applied to LBD to improve its performance. That is the purpose to which they are put throughout this work.

1.4 Thesis Outline

This thesis contains the following chapters. A brief synopsis of its contents accompanies each chapter entry listed here.

Chapter 1 introduces the field of LBD and justifies its relevance and importance. It highlights what the current open problems in the field are and gives an overview of some possible solutions which utilise recent advances in machine learning, particularly neural networks to improve biomedical NER, link prediction and the existing LBD paradigms.

Chapter 2 gives the relevant background needed to understand the rest of this thesis. It begins with a detailed overview of LBD; highlighting how it works, why it is a relevant research area and how it is evaluated. It looks at some of the approaches which have been taken to improve LBD. It then segues into the backgrounds of the methods investigated to improve LBD. It looks at NLP, in particular NER; link prediction; Multi-task Learning (MTL) and neural networks. Related relevant topics like Text Mining, knowledge discovery, representation learning, knowledgebases and graphs are also briefly covered for completeness.

Chapter 3 contains details of work to improve recognising biomedical concepts as entities in unstructured text, including using semantically-rich word embeddings for biomedical NLP along with using MTL and incorporating character-level features in deep neural architectures. It introduces each of those concepts then details how they are used to contribute to the improving of biomedical NER with experiments and results.

Chapter 4 contains details of work which uses neural networks to perform link prediction in large-scale biomedical graphs for various tasks, including LBD. It introduces the link prediction problem; motivates the use of neural approaches to this problem as well as informative evaluation techniques; and presents the experiments and results of that phase of the work.

Chapter 5 gives some background on the recently-released LION LBD system (Pyysalo et al., 2018) along with its LBD approaches and evaluation. It then details how neural network approaches and models, partly inspired by the link prediction approaches and models from Chapter 4 are applied to traditional LBD. It contains comparisons of the results from the proposed models and methods to those of the released system.

Chapter 6 concludes this thesis. It recapitulates the potential of LBD for use in modern biomedical research, the technical problems it still has and how they can be solved. It also deals with how the work proposed in the thesis can be solutions to those problems and the implications of the findings of the work. It then looks at possible directions in which this work can be taken in the near future.

1.5 Publications

In carrying out the work that this thesis presents, a few peer-reviewed publications were produced. A list of these and their respective brief overviews are given below.

1. **How to train good word embeddings for biomedical NLP**
   (*Chiu, B., Crichton, G., Korhonen, A., and Pyysalo, S. (2016a). How to train good word embeddings for biomedical NLP. ACL 2016, page 166.*)

Better inputs to neural network models can lead to improved results and has been a cause of recent state-of-the-art results in NER. I play a part in an effort to improve such inputs for neural networks for various biomedical tasks such as NER. My role is in the extrinsic evaluation of the inputs so in Section 3.3 we focus exclusively on that aspect of the work.

2. **Attending to characters in neural sequence labelling models**
(*Rei, M., Crichton, G., and Pyysalo, S. (2016). Attending to characters in neural sequence labeling models. In Proceedings of COLING 2016, pages 309–318.*)

   Since biomedical texts are sequential and biomedical entities encode much information at the character level, it makes sense to use a model which is designed to exploit both of these characteristics to improve performance. Long Short-Term Memory (LSTM) network is one such model so it is logical to use it for performing biomedical NER and incorporating character-level features. This gives rise to a role in work which utilises attention (a mechanism to improve neural network performance) in character-level LSTMs for several sequence labelling tasks including biomedical NER. The relevant parts of the work are described in Section 3.5.

3. **A neural network multi-task learning approach to biomedical Named Entity Recognition**
(*Crichton, G., Pyysalo, S., Chiu, B., and Korhonen, A. (2017). A neural network multi-task learning approach to biomedical Named Entity Recognition. BMC Bioinformatics, 18(1):368.*)

   In this paper, we investigate whether an MTL modelling framework implemented with a particular deep learning architecture (Convolutional Neural Networks) can be beneficially applied to biomedical NER. This is, to the best of our knowledge, the first application of this MTL framework to the task. Like other language processing tasks in biomedicine, NER is made challenging by the nature of biomedical texts which usually feature heavy use of terminology, complex co-referential links, and complex mapping from syntax to semantics. Additionally, the annotated datasets available for this task vary greatly in the nature of named entities (e.g. species vs. disease), the granularity of annotation, as well as in the specific domains they focus on (e.g. chemistry vs. anatomy). It is therefore an open question whether this task can benefit from MTL. Details are given in Section 3.4.

4. **Neural networks for link prediction in realistic biomedical graphs: a multi- dimensional evaluation of graph embedding-based approaches**

(*Crichton, G., Guo, Y., Pyysalo, S., and Korhonen, A. (2018). Neural networks for link prediction in realistic biomedical graphs: a multi-dimensional evaluation of graph embedding-based approaches. BMC Bioinformatics, 19(1):176.*)

Link prediction in biomedical graphs has several important applications including predicting Drug-Target Interactions (DTIs), Protein-Protein Interaction (PPI) prediction and LBD. It can be done using a classifier to output the probability of link formation between nodes. Recently several works have used neural networks to create node representations which allow rich inputs to neural classifiers. Preliminary works were done on this and report promising results. However they did not use realistic settings like time-slicing, evaluate performances with comprehensive metrics or explain when or why neural network methods outperform. We investigate how inputs from four node representation algorithms affect performance of a neural link predictor on random- and time-sliced biomedical graphs of real-world sizes (up to 6 million edges) containing information relevant to DTI, PPI and LBD. We compare the performance of the neural link predictor to those of established baselines and report performance across five metrics. This is described in Chapter 4.

5. **Neural networks for open and closed Literature-based Discovery**
(*Crichton, G., Baker, S., Guo, Y., and Korhonen, A. (Under Review). Neural networks for open and closed Literature-based Discovery.*)

Neural networks have demonstrated improved performance on LBD-related tasks but are yet to be applied to it. We propose four graph-based, neural network methods to perform open and closed LBD. We compare our methods with those used by the state-of-the-art LION LBD system on the same evaluations to replicate recently published findings in cancer biology. We also apply them to a time-sliced dataset of human-curated, peer-reviewed biological interactions. These evaluations and the metrics they employ represent performance on real-world knowledge advances and are thus robust indicators of approach efficacy. In the first experiments, our best methods perform 2-4 times better than the baselines in closed discovery and 2-3 times better in open discovery. In the second, our best methods perform almost 2 times better than the baselines in open discovery. These results are strong indications that neural LBD is potentially a very effective approach for generating new scientific discoveries from existing literature. Chapter 5 describes this work.

1.6 Conclusion

This Chapter explained what LBD is, why it can be particularly important in the biomedical domain and is presently a viable area of research. It presented arguments that its use is necessary to solve a glaring and growing problem in the field and postulates that the relevant computational technologies are at the stage where LBD can be implemented in a manner which gives useable output which can accelerate scientific discoveries. It also introduced link prediction as a powerful way of both performing LBD and transcending it and hinted how this can be done.

The overarching goal of my PhD was to investigate methodologies to improve the performance of tasks which produce input for LBD as well as LBD itself. Neural networks have shown their versatility and utility in their applicability to and improvement of various tasks, so using them as the primary engine to provide this improvement made sense. We used them in improving biomedical NER, improving link prediction on biomedical graphs and improving LBD with methods than can both be applied to the current dominant discovery paradigm and beyond it. This work was done within the context of the LION Project which is based on the ideas proposed in (Korhonen et al., 2014).

Chapter 2

Literature Review

2.1 Introduction

This Chapter gives the relevant background needed to understand the rest of this thesis. It begins with a detailed overview of Literature-Based Discovery (LBD) highlighting how it works, why it is a relevant research area and how it is evaluated. It then looks at some of the approaches which, like this work, have been taken to improve LBD. Following this, it segues into the backgrounds of the methods we investigate to improve LBD.

In general neural networks and deep learning are used throughout the work. Specific instances of their application are in the tasks of Natural Language Processing (NLP), in particular Named Entity Recognition (NER); link prediction and Multi-task Learning (MTL). Related relevant topics like Text Mining, Knowledge Discovery, representation learning, knowledgebases and graphs are also covered for completeness. Interspersed throughout are explanations of how these are applied to solve some of the pressing open problems in LBD and how they can perhaps be useful beyond the current dominant LBD paradigms.

2.2 Literature-based Discovery (LBD)

Literature-Based Discovery seeks to discover new knowledge from existing literature in an automated or semi-automated way (Henry and McInnes, 2017). LBD research has mostly focused so far on the biomedical domain, but it has also focused on other domains.

Within the biomedical domain it has been used to propose treatments for Parkinson's Disease, Multiple Sclerosis and cataracts (Kostoff, 2008b; Kostoff and Briggs, 2008; Kostoff et al., 2008a). It has also been used to propose new uses for curcumin and potential cancer treatments (Ahlers et al., 2007; Srinivasan and Libbus, 2004). However the areas which

have seen the most active research and results are in drug development and repurposing (Ahlers et al., 2007; Hristovski et al., 2010; Zhang et al., 2014) and in predicting adverse drug reactions (Banerjee et al., 2014; Shang et al., 2014). Outside the biomedical domain, proposals have focused on developing water purification systems, accelerating development of developing countries and identifying promising research collaborations (Gordon and Awad, 2008; Hristovski et al., 2015; Kostoff et al., 2008c).

Scientific literature is growing at an exponential rate (Hunter and Cohen, 2006) making it difficult for researchers to stay current in their discipline. This overwhelming volume of publications and the increasing necessity of researchers to specialise has led to non-interacting literature silos, which creates an environment where discoveries in one area are not known outside of it (Swanson, 1990a) and valuable logical connections between disparate bodies of knowledge remain unnoticed (Swanson, 1986b). This creates a situation where there is a real chance that pieces of information which can be combined to make breakthroughs are already discovered but are splintered and dispersed in the literature. LBD helps researchers to quickly gain information on relevant advances inside and outside of their respective niches and increase interdisciplinary information sharing. As the scientific literature grows, tools which aid researchers in finding and combining salient findings, like LBD systems, are becoming increasingly necessary for facilitating impactful research.

To deal with this challenge, LBD now uses various computational approaches and algorithms which seek to discover previously unknown associations or hidden links between pieces of existing knowledge by analysing literature (Smalheiser, 2012; Swanson, 2008). However the genesis of the field is generally attributed to Don Swanson's mostly manual discovery of the potential benefits of dietary fish oil for the treatment of Reynaud syndrome (Swanson, 1986a) by analysing their seemingly disparate literatures. This hypothesis was later shown to have some merit in a prospective study (DiGiacomo et al., 1989).

In that work, Swanson also introduced the idea of 'noninteracting literatures' and illustrated an example using Reynaud syndrome and dietary fish oil. Although there were almost 2,000 Reynaud papers and 1,000 fish oil papers, the literatures were isolated as determined by mentions or citations of works across the groups. An extensive search process revealed that only four papers mentioned papers in both groups, and of those four, two were co-incidental and the other two (which were similar enough to be counted as one) are review papers which mention both topics in separate sections. Sebastian et al. (2015) illustrated this disconnect between those same literatures at that time by comparing a figure of the co-citation links between the literatures prior to Swanson's 1986 publication and after it (Figure 1 of that work); there is a stark difference in the amount of links across the literatures. Since Swanson's work, the concept has been used in LBD works under different names; for example Katukuri

et al. (2012) speaks of 'cross-silo' literature hypotheses, Thaicharoen et al. (2009) of 'disjoint sets of literatures' and Hu et al. (2006) of 'complementary and non-interactive biomedical literature'.

2.2.1 Categorising LBD Approaches

Swanson (1986a) defined the basic and most dominant type of LBD in the literature, called the ABC paradigm because it centres around three concepts referred to as A, B and C. The main idea is that if there is a connection between A and B and one between B and C then there is one between A and C which, if not explicitly stated is yet to be explored. Within the ABC paradigm, there are two types: *open and closed discovery*. In open discovery (also called hypothesis generation), only A is given. The system finds Bs (called intermediate or linking terms) and uses them to return possibly interesting Cs (called candidate or target terms) to the user, thus *generating* hypotheses from A. With closed discovery (also called hypothesis testing), the A and C are given to the system which seeks to find the Bs which can link the two, thus *testing* a hypothesis about A and C. From this starting point, several related approaches have arisen.

Sebastian et al. (2017a) distinguishes between the traditional approaches which are characterised as statistical, knowledge-based and visualization; and newer paradigms which they term 'emergent' and believe will define the field in the future. They see these as characterised by two main trends: 1) integrating traditional approaches such as statistical, knowledge-based and visualization approaches to create a unified LBD solution and 2) using techniques borrowed from other research fields including link prediction and machine learning which offer a different angle on how LBD can be performed and its problems addressed.

Henry and McInnes (2017) prefer to delineate approaches based on how they represent terms, what types of relationships they use and how they find linking and target terms. Using this rubric, they categorise LBD models as co-occurrence, semantic or distributional.

There are different ways of categorising LBD systems and most categories used would only capture some aspect of an approach while appearing to ignore others. In general, the myriad papers spawned since 1986 on the topic would defy simple categorising as most would span several categories. Thus the categories used here simply refer to what may be prominent features of the approaches mentioned within them and are to be read as such. In general a method explained under a particular category would also employ elements applicable to other categories and these should be obvious; where appropriate we point out when they may not be.

2.2.2 Representing Concepts

The way an LBD approach represents entities affects which entities it would and would not capture from text, how it will treat synonymous entities and other things which would affect the discovery phase. There are three main approaches to representing entities about which knowledge is discovered in LBD. They have been represented as words or terms; concepts and keywords. Each of these entail a precision-recall trade-off; for example words/terms tend to have high recall but low precision.

Words or terms: These approaches represent concepts as words as they are found in text. This is the easiest way of representing entities, so it is unsurprising that it is the most popular approach used when the field was nascent (Swanson, 1986a, 1988).

Knowledgebase entries: These approaches represent concepts as entries from a external resources such as United Medical Language System (UMLS) (Bodenreider, 2004).

Weeber et al. (2000) developed an NLP system called DAD (Drug-Adverse Drug Reaction-Disease) to assist biomedical experts in generating and testing hypotheses, mainly for drug discovery research. They were able to use concepts by mapping words in titles and abstracts to entries in the UMLS Metathesaurus. By using such entries they were able to circumvent some of the problems involved in using terms such as the difficulty of identifying them in unstructured text, the need to use lists of stop words and complex approaches to capture synonyms and variants. Using concepts also facilitated the extraction of multi-word entities and filtering the amount of entries the system had to deal with by using filters constructed from UMLS semantic types. To demonstrate the usefulness of their discovery system, Weeber et al. (2003) published the results of a study on potentially new targets for the drug Thalidomide - a withdrawn sedative. They found evidence in PubMed suggesting that Thalidomide could be an effective treatment for several conditions including chronic Hepatitis C and acute Pancreatitis.

Keywords: These approaches represent concepts as keywords, which are less restricted than knowledgebase entries but more so than words. They can include Medical Subject Headings (MeSH) descriptors (Lipscomb, 2000); a controlled vocabulary thesaurus used for indexing articles in MEDLINE and PubMed.

Stegmann and Grohmann (2003) analysed the strength of co-occurrence for pairs of keywords assigned to MEDLINE documents. Keywords included MeSH, Enzyme Commission Numbers and Chemical Abstracts Service Registry Numbers. The analyses lead to maps or "strategical diagrams" of clusters containing keywords. Promising terms linking complementary but disjoint literatures tend to appear in regions of low centrality and density. They validated their approach by replicating Swanson's Raynaud's syndrome - fish oil

and migraine - magnesium findings. They also found evidence for a relationship between manganese, prions and neurodegenerative diseases.

2.2.3 Use of (non-distributional) Semantics

These approaches made use of tools which seek to capture and exploit the semantics in text to facilitate LBD. The hope is that using semantics will allow the approach to capture a deeper meaning of the information in the text which is not captured by simple term matching etc. in text.

Srinivasan (2004) used UMLS semantic types and MEDLINE metadata (MeSH terms) in Text Mining algorithms for discovery. This was done by building profiles of research topics based on weighted MeSH terms from MEDLINE documents, where weights are estimated within semantic types based on the weighted terms it contains. Taken together, weighted terms constitute a profile of the topic of interest. Topics for profiling can be single words or phrases that need not be composed of MeSH terms. The method was evaluated on the first step of the discovery process: the identification and ranking of key terms on five of the discoveries proposed by Swanson. They found that across all the discoveries (in open and closed discovery settings) the method was able to rank key terms highly within semantic types.

Hristovski et al. (2006) highlights two main problems with using simple concept co-occurrence as the primary relationship between entities for LBD: 1) It provides no semantic information about the nature of the relation between the entailing concepts which leads to an inability to produce explicit explanations of the discovered relations and, 2) not all co-occurrences are indicative of useful relations, so systems employing this approach tend to produce large numbers of spurious relations which in turn mean that users must read large numbers of papers when reviewing candidate relations. They address these deficiencies with the use of semantic relations to augment co-occurrence processing. To achieve this, they combine the output of two NLP systems (SemRep (Rindflesch and Fiszman, 2003) and BioMedLee (Lussier and Friedman, 2007)) to provide *semantic predications*. Semantic predications are subject-relation-object triples extracted from text and have since been widely used in LBD. In order to make use of semantic predications in LBD, they introduce the notion of a *discovery pattern*, which contains a set of conditions to be satisfied for the discovery of new relations between concepts. The conditions are combinations of relations between concepts extracted from MEDLINE citations. They evaluated their approach on two proposed discovery patterns using the BITOLA LBD system (Hristovski et al., 2001) to replicate Swanson's Reynaud syndrome discovery and by uncovering links between Huntingdon's disease. For the latter, they sought links between the disease and changes

in substances or body functions which could be potential therapeutic targets leading to the development of new treatments. They found that their approach produced a smaller number of false positives while facilitating user review of new relations.

Cohen et al. (2012) sought to automate the identification of these discovery patterns using what they termed *Predication-based Semantic Indexing (PSI)*. These patterns were derived from semantic predications extracted from biomedical literature using SemRep and used to direct the search for known treatments for a held-out set of diseases. PSI represents both concepts and the relationships between them as vectors in hyperdimensional space. They exploited the geometry of this space to use reversible vector transformations to perform inference for LBD. They compared their method to the Reflective Random Index (RRI) (Cohen et al., 2010). PSI was further used in (Cohen et al., 2014), along with SemMedDB, a publicly available database of semantic predications provided by SemRep for query expansion in semantic search. Their approach was able to rediscover discovery patterns that were constructed manually in previous work. It was also able to find a set of previously unrecognised patterns. They propose that the method results in better recovery of therapeutic relationships than with models based on distributional statistics alone.

Goodwin et al. (2012) used the mathematical models underlying Information Foraging Theory (IFT) which predict information foraging behaviours of users on the web to design a discovery browsing system which performs biomedical discovery. The system mines a semantic network created by millions of semantic predications from several million MEDLINE citations using SemRep. They evaluated the system by giving it some seed terms and testing its ability to use them as starting points to replicate two previous discoveries linking testosterone to sleep and sleep to depression. The tests demonstrated that the system was able to predict concepts that were determined as playing a role in novel proposed hypotheses. This work shares similarities to the field of concept-based Exploratory Search (ES) (Crichton, 2013; Marchionini, 2006; White and Roth, 2009).

Even closer to the field of ES is the *discovery browsing* work of Wilkowski et al. (2011) which aimed to extend LBD methodology beyond making discoveries to a principled way of navigating through aspects of a particular research area and to reveal crucial relationships in the domain in response to a query instead of merely document retrieval. These relationships allow a user to evolve their query as they continue to use the method. To achieve this, they incorporate semantic predications and graph-based methods in order to guide researchers through the relevant literature on a user-specified biomedical phenomenon. Their methodology included creating a graph of the relevant predications, extracting paths from the graph and ranking them, and manually inspecting a small subgraph based on selected paths. At several steps in the process, the system's output is influenced by the user's interaction.

They demonstrated their approach by using it to investigate aspects of depressive disorder especially as it pertains to the interaction of inflammation, circadian phenomena, and the neurotransmitter Norepinephrine; and deemed it up to the task of doing so.

Preiss et al. (2015) introduced an LBD system that used matrices to encode weighted relations between terms. They explored six relations: three based on co-occurrence and three on semantic linguistic analysis. They created an adjacency matrix of the graph formed from the relationships between terms in a corpus: positive integers if a relation is detected; zeroes otherwise. Hidden knowledge is then found by looking for non-zero terms in the matrix generated by the difference of the norms of this matrix and its square. They evaluate the system by using seven discoveries that have previously been used for replication experiments in LBD as well as by time-slicing a database to create three different gold standards which depend on how many of the discoveries were identified in the evaluation slice by one, two or all of the same three semantic approaches used. They found that approaches that use relations extracted through automatic linguistic analysis reported several orders of magnitude fewer instances of hidden knowledge than approaches that use term co-occurrence relations. This drastic decrease did not have an adverse effect on the system's ability to replicate existing discoveries in most cases. They thus concluded that using automated linguistic analysis in relation identification for LBD provides significant benefits.

2.2.4 Distributional and Statistical Approaches

These approaches take advantage of distributional and/or statistical methods to perform LBD. They usually apply the methods to extract information from texts then use that information for the task.

Gordon and Dumais (1998) used Latent Semantic Indexing (LSI) (Deerwester et al., 1990) for improving information retrieval effectiveness in general and LBD in particular. They proposed that their method could aid in either of the two phases of LBD: during the search for intermediate literatures and in helping to identify potential discovery literatures. In the latter, LSI can be used in two ways: to locate and retrieve a set of documents associated with suspected intermediate literature, or by analysing the wider literature in which the terms of interest in suspected connections occur. Following Gordon and Lindsay (1996), they evaluated the approach by seeing how well it performed on two of Swanson's discoveries on those tasks. This evaluation suggested that LSI might be a useful tool in LBD because it provides another technique that can be considered in uncovering hidden discoveries as other methods of performing LBD may fail when applied to certain problems.

Lindsay and Gordon (1999) used four lexical statistics (token frequency, document frequency, relative frequency compared to entire corpus, and term frequency-inverse document

frequency) to perform LBD on MEDLINE documents. They determined connections between entities using uni-, bi- and tri-grams. They evaluated their method by testing to see if it would retrieve any of the intermediate literatures of Swanson's migraine-magnesium hypothesis. Their automated methods were able to identify 10 of 12 known intermediate topics relating migraine and magnesium within a "reasonably sized" candidate list.

Wren (2004) used Mutual Information Measure as a proof of principle that statistical methods of association information can be extended to implicit relationships and thus be used to uncover implicit associations in text. Such inferred associations can be used for LBD. To evaluate his approach, fifty objects of research interest (e.g. biomedical entities or ontology categories) were chosen at random from MEDLINE and random word databases and used to create a network of associations. Each object was analysed to identify and rank other objects that shared relationships with it as described and the Area Under the Curve (AUC) was taken for four ranking methods and a random baseline. He also performed a detailed case study to uncover and rank inferred relationships to Capsaicin as well as re-evaluated two of Swanson's hypotheses. He concluded that MIM can be effectively extended to relationships which are not directly observable. The shared minimum MIM (MMIM) model was found to perform best using observed strength and frequency of known associations as the metric.

Symonds et al. (2014) carried out a survey comparison on the complexity of four corpus-based distributional approaches for LBD: Latent Semantic Analysis/Indexing (LSA), Hyperspace Analogue to Language (HAL), Random Indexing (RI) and Tensor Encoding (TE). In general these models work by constructing matrices whose values represent distributional semantics about the terms and/or documents in large corpora, reducing them (using for example SVD) so that only the most relevant information remains then calculating similarity metrics (e.g. cosine similarity) on the vectors of the terms of interest (A-C terms) for LBD. They found that models which store representations in fixed dimensions provide superior efficiency on LBD tasks. In particular, the TE model was well-adapted to the task of open discovery due to its ability to complete the steps of storing representations and computing similarities from them in a single step.

2.2.5 Graph-based Approaches

The nature of LBD lends itself easily to being performed by mining a graph containing the requisite knowledge. It is no surprise then that several approaches have taken a primarily graph-based approach to LBD and others have used it in conjunction with other approaches.

Katukuri et al. (2012) modelled the existing biomedical literature as a network of biomedical concepts using concept co-occurrence in documents. They then performed supervised link prediction on the graph to generate hypotheses. They manually created two topological

and two semantic features which were then used to create input to Decision Trees and a Support Vector Machine (SVM) to classify whether a link exists between future nodes or not. The links predicted represented hypotheses from non-interacting fields. They evaluated their approach by time-slicing the graph and predicting links in the later time period. They found an improvement of 7-9% in classification accuracy when adding semantic type and author -based features.

Eronen et al. (2012) presented Biomine which performed link prediction on a biomedical graph for general knowledge discovery. The weighted, heterogeneous graph was constructed by combining data from several biomedical databases including PubMed, Entrez Gene (Maglott et al., 2005), Gene Ontology (Ashburner et al., 2000) and UniProt (Apweiler et al., 2004). They performed the linking using proximity measures including probability of best path, network reliability and expected reliable distance. It was evaluated using time-slicing to determine which links were actually added to the various data sources at a later time period. They used the Area Under the ROC Curve (AUC-ROC) to quantify this. They found that combining data from several sources was beneficial to the task and that the proximity measure which used random walks on the graph was the best.

Ding et al. (2013) proposed the idea of *entitymetrics* which they define as using knowledge units as entities to measure impact, knowledge usage and knowledge transfer to facilitate knowledge discovery. They constructed a bio-entity citation network among biomedical papers which captures information on how entities provide signals for citation relationships. They then used macro, micro and meso-levels features of the graph to calculate scores for nodes of interest. To evaluate their approach, they constructed such a graph for the drug Metformin which was developed to treat Type 2 diabetes but has also shown potential to prevent cancer, obesity, depression and ageing; making it a drug of high interest for biomedical knowledge discovery. After calculating the scores using the features, they compared it with a score provided by the dataset which indicated likelihood of linkage in a network reserved for inference. Their method proved to be useful but it was only evaluated with a single entity and the features used on the graph were selected manually, although they are general graph-theoretic features.

Cameron et al. (2013) sought to recover and decompose Swanson's dietary fish oils – Reynaud syndrome hypothesis into its constituent high level and atomic parts, which they refer to as primary, secondary and association hypotheses. They extracted semantic predications (subject, predicate, object triples in this case) from the biomedical texts which Swanson used, then used these predications to create a predications graph which was used to generate semantic associations. These were then used to manually create subgraphs of the associations expressed by Swanson. Background knowledge and domain expertise were used

to improve the subgraph creation. They found that they were able to recover and decompose well with titles, abstracts and full texts but had almost no success with simply titles and abstracts. The subgraphs provided several insights into unmentioned associations which could explain why fish oil affected Reynaud syndrome and thus gives some explanation and interpretability. There were several limitations of this work: they evaluated on only a single case; their approach relies heavily on external resources, tools and pre-defined semantic predications; and the subgraphs were manually constructed by persons with domain knowledge. They automated the latter procedure in later work (Cameron et al., 2015). In that work, they looked at associations along different dimensions such as Cellular Activity and Pharmaceuticals. MeSH descriptors of documents were used to provide context to the algorithm for determining which paths to combine into subgraphs using Hierarchical Agglomerative Clustering (HAC). Their results showed that this context was more important to elucidating hidden connections than frequency, connectivity or specificity in graphs. This work was evaluated on nine cases; three in great detail and six in less.

Sebastian et al. (2015) performed LBD as link prediction between disjoint research areas over a heterogeneous graph of bibliographic information. They created paths on heterogeneous graphs containing information such as terms, author, publisher, topic, cited-reference and citing-paper as vertices and relationships such as cites, cited_by, published_by etc. They evaluated by testing the ability of their approach to predict future co-citation links for Swanson's fish oil and Reynaud syndrome proposal. This was done by framing links as three classes (inter-cluster, within-cluster, and no-links) and using several machine learning classifiers to classify a link into one of those classes based on input from the meta-path features. They downplayed the use of the content of papers to predict new connections between complementary but non-interacting fields of research which is the aim of LBD, instead arguing that the bibliographic information on a heterogeneous graph is quite good by itself. Their approach was highly manual: the meta-path features used were manually constructed and methods for scoring the association strengths of the paths were selected manually which left them susceptible to bias. In (Sebastian et al., 2017b) they expanded on that work by evaluating on an additional Swanson case (Magnesium-Migraine) and adding semantic processing in the form of word sense disambiguation and topic modelling to the approach.

Kastrin et al. (2016), building on earlier work (Kastrin et al., 2014) which showed the plausibility of using link prediction for LBD, fashioned LBD as a classification problem on a graph of MeSH terms. They used unsupervised and supervised link prediction methods on this graph to predict previously unknown connections between the biomedical concepts represented by the MeSH terms. Their unsupervised approaches were Adamic-Adar (AA)

(which was the best performer), Common Neighbours (CN), Jaccard Coefficient (JC), and Preferential Attachment (PA). For the supervised learning approach, they sought to decide whether those proximity measures can be combined to define a model of link formation across all four predictors by applying Decision Trees, *k*-nearest neighbours, logistic regression, multi-layer perceptron, naïve Bayes, and random forests (which was the best performer). They concluded that the supervised statistical learning approaches outperformed the unsupervised approaches using AUC (ROC) as the metric.

Building on their existing LBD system (Preiss et al., 2015), which used some graph-based approaches, Preiss et al. (2018) created the publicly available HiDE (Hidden Discovery Explorer). HiDE is an online knowledge browsing tool which allows fast access to hidden knowledge generated from abstracts in MEDLINE. It also allows users to explore the full range of hidden connections generated by an underlying LBD system. It combines a graph-based approach which allows hidden knowledge to be generated on a large scale with an inference algorithm to identify the most promising (i.e. most likely to be non-trivial) information. It used graphs in knowledgebase completion which generates new connections by performing random walks through a graph of the knowledgebase. It also uses linguistically-motivated subject-relation-object triples (such as X-treats-Y or X-affects-Y) extracted from a SemRep-annotated version of PubMed. HiDE then generates a list of potentially relevant UMLS Concept Unique Identifiers (CUIs) for the user to select one and the hidden knowledge proposed is grouped by MeSH terms. To illustrate the system's utility, they presented an image of the tool's output when replicating the connection between Swanson's Raynaud syndrome and fish oil discovery from MEDLINE publications in the period 1960-1968. The image showed the hidden knowledge generated by entering the search term *raynaud* and that the link to *fish oil* was found.

2.2.6 Miscellaneous

Swanson and Smalheiser (1997) introduced interactive software and database search strategies (ARROWSMITH) to facilitate the discovery of unknown cross-specialty scientific information. Using these strategies, a user begins by searching MEDLINE for article titles that identify a topic of interest. The software then uses the titles of the retrieved papers to create input for additional searches and produces a series of heuristic aids that help the user select a second set of articles which are from a complementary research area. Two sets are considered complementary if taken together they can reveal new information that cannot be inferred from either set alone. The software also helps the user identify new information and derive from it novel testable hypotheses. Evaluation consisted of testing whether the system can at least be helpful in rediscovering complementary structures in three completed

and published analyses of complementary non-interactive literatures which were three of Swanson's discoveries. It additionally evaluated the performance on a usage variant, which was focused on elucidating intermediate pathways or mechanisms by which two concepts of interest are linked. To perform the latter it used a further three cases: Indomethacin and Alzheimer's disease, Estrogen and Alzheimer's disease and Phospholipases and Sleep. In two of the cases, suggestions from the system were brought to the attention of relevant researchers for investigation and in the other case, a link was published while the manuscript was under review.

2.2.7 Summary

In summary, there has been a plethora of work on LBD since it was proposed as a strategy for discovering new scientific knowledge. These works have proposed several different approaches to LBD even when they used some similar components. For the most part the processes they have used to identify biomedical entities in text mostly consists of using n-grams in text and of matching to a list of known terms from entries in external resources. That makes them reliant on these resources being complete and updated; and makes them as error-prone as the simple string matching process for information extraction can be. There is scant evidence that the discovery methods proposed by these systems can scale to produce a wide range of good quality discoveries.

One of the strengths of neural networks is their theoretical ability to incorporate several of the beneficial strategies used by existing models presented here. For example, neural models which process text make use of word embeddings produced by other neural networks which are vector representations that encode distributional semantics and capture various linguistic semantic properties. While these methods would still miss information which is pertinent to LBD such as negation, hedging and compositional concepts, they are able to provide more information to neural models which they are used in to improve performance.

Building on the success of such work, such embeddings have also been used to encode the information in graphs making neural processing of them for a wide-range of tasks, including LBD, possible. Similar to the embeddings created by text, the graph embeddings are also expected to provide richer inputs to neural models but can also potentially have shortcomings including ignoring edge weights and direction.

2.3 Evaluating LBD Systems

Evaluation is needed to determine which methods and models are successful as well as to quantify any successes. This has proven to be difficult in LBD thus far for several reasons which include: disagreement about the role of LBD systems in research (i.e. the line between aiding and replacing researchers), and thus what makes a successful one; difficulty in determining metrics for how useful, interesting or actionable a discovery is; and difficulty in objectively defining a 'discovery', which has led to difficulties in creating a standard evaluation set which quantifies when a discovery has been replicated or found.

As a result, several approaches have been used in the literature to evaluate a proposed approach. The more prominent ones are covered here along with some of their strengths and weaknesses.

2.3.1 Replication of Previous Discoveries

One of the more popular methods used in LBD is to seek to replicate previous discoveries (Gordon and Lindsay, 1996; Stegmann and Grohmann, 2003; Swanson and Smalheiser, 1997; Weeber et al., 2000). Discovery replication consists of replicating a prior discovery made by actual researchers or previous LBD systems. These have usually been LBD-based discoveries as they are relatively easy to quantify as opposed to other discoveries. This means that there are only a handful of such discoveries to use and there is a danger of designing systems which are tuned to perform well on these discoveries but are not generalisable; so other evaluation techniques should be used in conjunction with this approach. Additionally, this technique ignores all other discoveries the system may propose, which may be as equally valid as the ones being replicated. Despite its shortcomings, discovery replication seems a reasonable necessary first hurdle for a proposed system to get past, and it is the only evaluation found in most older work on the topic.

In this type of evaluation, the state of the literature before the discovery to be replicated is used to generate discovery candidates in the form of the target term. Success is determined by whether the term of interest is returned as part of this list or not. However, the presence of the desired term in a list of target terms is insufficient to indicate the likelihood of the term being noticed by a hypothetical researcher. This approach also does not allow for quantitative comparisons between systems or system components.

For these reasons, it is now standard procedure to perform LBD as a ranking task and to report the rank of the term(s) of interest. The higher the rank of the term(s) of interest, the better the system. These techniques can be used to evaluate both open and closed discovery systems depending on whether the ranked list consist of linking or target terms.

2.3.2 Time or Literature Slicing

This method consists of splitting the existing literature at a certain point in time. The system is then trained on the literature from the period before the split with the aim of determining how many of the discoveries in the latter period it can discover. Issues with this approach include: the unclear definition of a discovery, as mentioned before, leading to some ambiguity about what the gold standard for the evaluation should be; interpreting the results in light of the metrics used to measure performance; and not being able to determine if a proposed discovery is incorrect or simply has not been made as yet.

In the absence of a perfect list of the evaluation gold standard, this approach estimates it by finding instances of the defined relationships in the test set which are not in the training set and can be reasonably inferred from it. This means that the evaluation depends heavily on what constitutes a relationship for the given system. If a noisy relationship like co-occurrence is used, then the evaluation will be noisy (and easy to perform well on). Existing systems have used term co-occurrences (Hristovski et al., 2000), relationships from external biomedical resources (e.g SemMedDB) (Cameron et al., 2015) and semantic relationships (Preiss et al., 2015). A high precision approach would be to get expert opinion to generate a list of gold standard terms (Yetisgen-Yildiz and Pratt, 2009), although this is expected to be time-consuming, possibly expensive and have low recall rates.

One example of using expert opinion in this manner is the recent LION LBD tool (Pyysalo et al., 2018). In addition to replicating five of Swanson's discoveries, they evaluated on a Cancer Discovery dataset created by cancer researchers which can be used as a gold standard for LBD evaluation in biomedical domain. It features A, B and C triples - extracted from publications in top biomedical journals - and the year of the publication that made the connection. Systems are evaluated on how highly they are able to rank the AC connection when trained on PubMed documents released before the AC publication year. While this method employs the literature slicing strategy, it differs in that it provides a fixed set of certified discoveries curated by domain experts and supports testing of a system's ability to discover multiple intermediary links which point to AC.

The advantage of this evaluation approach is that it tends to produce an indicator of the system's performance on a large number of possible test instances. This gives rise to the need for evaluation metrics which can quantify the system's performance on large, ranked lists. For this, LBD works usually use metrics more popular in Information Retrieval (Yetisgen-Yildiz and Pratt, 2008). These include Precision, Recall, Precision-Recall Curve, Receiver Operator Characteristics (ROC) Curve, Precision at K, Mean Average Precision (MAP) and F-score. More details on these are given in Section 2.3.4.

2.3.3 Proposing New Discoveries

Proposing new discoveries or treatments goes a step beyond replicating past discoveries or predicting instances of a particular relationship after a point in time and shows that a system is capable of being used in realistic situations and has been used in several works (Hristovski et al., 2010; Stegmann and Grohmann, 2003; Swanson and Smalheiser, 1997; Wren et al., 2004). To be valid, this is usually accompanied by vetting of the proposal by a recognised domain expert or peer-reviewed publication in the relevant domain. An example of this is that Wren et al. (2004) identified compounds which their system predicted might affect the development and/or progression of cardiac hypertrophy and performed laboratory tests in a rodent model and found that the compound Chlorpromazine reduced the progression of cardiac hypertrophy.

In summary, there have been several approaches to evaluating LBD approaches intrinsically and extrinsically. There has been heavy use of seeking to replicate specific discoveries, especially those proposed by Swanson and others usually with a time-slicing element although more generalisable evaluations have been used. Unfortunately, in general very few evaluations can lay claim to demonstrating that an approach is generalisable and practically useful. In the end however, the ultimate extrinsic evaluation of an LBD system is still its uptake by practitioners and users. This has been and still is a problem for LBD (Kostoff, 2008a).

2.3.4 Evaluation Metrics

The evaluation metric that is used is an important aspect of LBD system evaluation because it highlights what areas of the task are being done well and which are not. This is dependent on which of the above evaluation methods is chosen, because that influences how many evaluation cases there are and thus what metrics are suitable and even possible. As mentioned above, it is now expected that terms of interest and linking terms will be ranked in the returned list.

In cases where there is a single or only a few correct (gold) terms in the evaluation set, Rank (single gold) or Mean Rank (multiple golds) makes sense. When there are many possible correct terms in the evaluation set, as mentioned above, LBD works usually use metrics more popular in Information Retrieval. These include Precision, Recall, Precision-Recall Curve, Receiver Operator Characteristics (ROC) Curve, Precision at k, Mean Average Precision (MAP) and F-score. Other metrics which are suited to ranked lists include Mean Reciprocal Rank (MRR) and Mean Relevance Precision (Mean R-precision). Some of these

metrics are dealt with in more detail later in the thesis (Section 4.6.1 and Appendix C), but for completeness we give brief descriptions of each here.

Area under the Precision-Recall Curve (AUC-PR): *Recall* measures what percentage of all possible positives in the evaluation set were returned by the system. *Precision* measures what percentage of the results returned by the system are true positives. These metrics are used to construct a Precision-Recall Curve which illustrates how the increase in recall affects precision.

Area Under the Receiver Operating Characteristics Curve (AUC-ROC): *True positive rate* is equivalent to recall. The *fallout* or *false positive rate* measures how many negatives were returned as false positives by the system. These metrics are used to construct a Receiver Operating Characteristics (ROC) Curve which illustrates this relationship.

Precision at *k*: Other metrics measure performance across all recall levels but most applications with ranked results are only interested in the quality of highly ranked results. *Precision at k* or the *top k predictive rate* is the percentage of true positives among only the top k ranked results.

Mean Average Precision (MAP): Given a ranked list of results relevant to a particular query or term, we can calculate the precision at each true positive. The average of these values gives the average precision for that query or term. This done over all queries/terms in an evaluation set gives a single-value measure of a system's performance across all queries/terms.

Averaged R(elevant)-Precision: Similar to MAP but instead of calculating the precision after each positive in the list of results for a given query or term, precision is only calculated with the top R results. R is determined by how many true positives exist for the query or term. This metric is similar to precision at k except that instead of having a fixed k, it changes based on the amount of positives each query has so that a query with less than k positives is not unfairly penalised and a query with a lot more positives than k is not trivially easier for the system to perform well on.

Mean Rank (MR): This is the mean of the ranks of the correct or positive terms in the list of ranked results returned.

Mean Reciprocal Rank (MRR): The MR is not normalised across lists of results of varying sizes, which makes it susceptible to distorted results from relatively long or short lists. It also does not give strong prominence to systems which rank correct terms at the top of the list which is of importance for ranking tasks. By inverting the ranks, both of these issues are solved.

2.4 Improving LBD

There have been numerous approaches aimed at improving various aspects of LBD with a view to improving its performance. These include: using machine learning to rank intermediate/linking terms for relevance and predict them for a given search (Torvik and Smalheiser, 2007); removing the need to generate and evaluate intermediate terms (Cohen et al., 2010; Gordon and Dumais, 1998); pruning the amount of intermediate terms by using knowledgebases (Srinivasan, 2004; Weeber et al., 2001); making intermediate connections more transparent (Cameron et al., 2013; Weeber et al., 2001); use meta information (e.g. bibliographic links) to make discoveries (Sebastian et al., 2017b); combining disparate information sources to obtain better information (e.g. bibliographic, knowledgebases, text) (Ding et al., 2013; Eronen et al., 2012; Kostoff, 2010; Sebastian et al., 2015); expanding queries beyond the core terms (Cameron et al., 2015; Kostoff et al., 2008b; Wilkowski et al., 2011); using Word Sense Disambiguation to resolve ambiguities pervasive in biomedical texts (Preiss and Stevenson, 2016); using semantic information to reduce noise in inputs (Preiss et al., 2015, 2012) and filtering the linking or resulting terms (Hristovski et al., 2003; Preiss and Stevenson, 2017).

Of these methods, some categories are of particular interest to this work: works which used machine learning (ML); works which included utilising graphs and those which used NLP methodologies. We have already covered most of the list from the previous paragraph in detail in Section 2.2. Here we will add information on a few methods taken specifically to improve LBD and highlight which works are relevant because they are the closest approaches to what this thesis deals with. Relevant ML works include (Kastrin et al., 2016; Katukuri et al., 2012; Sebastian et al., 2015). Relevant works which utilise graphs were dealt with in Section 2.2.5; of particular interest are (Cameron et al., 2013; Eronen et al., 2012; Katukuri et al., 2012). Works which make use of NLP methodologies include (Preiss et al., 2015; Sebastian et al., 2017b).

Hristovski et al. (2003) integrated background knowledge into an LBD system aimed at discovering candidate genes for diseases. The knowledge was the chromosomal locations of the diseases and genes from external resources such as LocusLink (Pruitt and Maglott, 2001) and Online Mendelian Inheritance in Man (OMIM) (Hamosh et al., 2005). When given a disease as the starting point for LBD, this allowed the system to constrain proposed candidate genes to be in the same location as the disease.

Gulec et al. (2010) investigated the effectiveness of pruning and grouping, two approaches used to improve the performance of LBD systems. Pruning refers to removing very general (thus uninformative) terms, terms which are highly related to the initial term and using semantic types to eliminate conflicting or impossible relationships. They found that grouping

on the target term led to decreased performance; that pruning terms which are closely related to the starting term did not have an impact; and that semantic pruning had a negative impact.

Preiss et al. (2012), like Hristovski et al. (2006) before, bemoan the use of simplistic approaches to obtaining information from biomedical texts for LBD, such as bag-of-words for entity extraction and co-occurrence for relationships. They list some of the problems these can lead to and some possible solutions. The problems include ambiguity among biomedical terms which can lead to real connections being missed and false connections being made; relationship types based mostly on textual co-occurrences which give no explanation of how the entities are related (if at all) and may miss linguistically important phenomena such as negation; and lack of interpretation of the results proposed by the system which is of little help to persons who require an explanation for the conclusions reached by the system (especially in light of a non-trivial possibility of incorrect conclusions), which applies to most, if not all, categories of LBD users. They propose Word Sense Disambiguation (WSD), Information Extraction (IE) and data mining techniques respectively as solutions to these issues.

Following on from the first issue raised in that work, Preiss and Stevenson (2016) detailed how ambiguity among biomedical terms is a potential problem for LBD, leading to abundant low-quality proposals. To study this, they integrated WSD into an LBD system (Preiss et al., 2015) using three WSD approaches of varying performance levels. They found that the performance of the LBD system improved as the WSD approach's performance did, which was a strong indication that LBD performance benefits notably from WSD.

Rastegar-Mojarad et al. (2016) used semantic predications, from SemMedDB, of drug-gene and gene-disease relations to perform LBD. They referred to these predications as 'causal findings' and investigated the use of the sentences from which the predications were extracted as context to classify the relationships into desired classes for the tasks of drug repositioning, identifying adverse drug events and detecting drug-disease relationships. They extracted features from the sentences and used them to train classifiers (including SVM, Naïve Bayes and random forests) to perform the classification.

Given the high amount of new knowledge proposals that LBD systems generate, approaches to filtering out the high-quality ones as a means of improving LBD have been a research focus over the years. These approaches have included: synonym merging (Cameron et al., 2015), using stoplists (Swanson and Smalheiser, 1997; Swanson et al., 2006), literature reduction (Swanson et al., 2006), restrictions based on term or semantic type (Hristovski et al., 2003; Yetisgen-Yildiz and Pratt, 2008), confidence thresholds (Hristovski et al., 2003) and using concepts and keywords instead of terms (Weeber et al., 2001).

Preiss and Stevenson (2017) performed an extensive study of the quality of the proposals of an LBD system and investigated the efficacy of four filtering approaches. The approaches they investigated were synonym merging, semantic type restriction, using a common linking terms stoplist and identifying and removing connections of common linking terms. They evaluated the performance of a system which incorporated these methods on replicating seven previously proposed discoveries as well as on a time-sliced dataset using SemRep relations as the relationship between the entities. They found breaking the connections between common linking terms to be the most effective approach.

2.5 Natural Language Processing (NLP) and Text Mining (TM)

LBD is a secondary process in the sense that it uses the outputs of other processes as its inputs. The main inputs required are concepts (or entities) and relationships between those concepts. Current methods of procuring concepts include dictionary matching in text and use of biomedical knowledgebases (such as MeSH and SemMedDB). These methods require resources and humans for maintenance and are prone to becoming outdated. They also restrict recall to the concepts found within them, although precision tends to be high when using them.

Text Mining (TM) is a general approach which aims to automatically identify, extract and discover new information from text by combining approaches from IR, NLP and data mining. A related field of TM that is pertinent to this thesis is NLP, which is concerned with the interactions between machines and natural languages, particularly how to enable machines to process large amounts of natural language data. These offer an alternative way of procuring concepts and relationships for LBD as they can process the myriad of biomedical text available in a manner which can increase recall and require little or no resources to maintain their flexibility to identify concepts and relationships.

Biomedical TM has become increasingly popular over the past decade or so in response to the exponential growth of biomedical scientific publications. Resources and NLP techniques such as part-of-speech (POS) tagging and parsing have been developed for biomedicine. IR and Information Extraction (IE) are now developed, and relatively accurate techniques are now available for recognising biomedical named entities (NER), relations and events in text (Korhonen et al., 2014).

NLP has made great strides in recent years due to the advent of neural networks and improved inputs into neural network models which perform NLP tasks. Because of this, these

are two areas of active research in the field. Given these recent advances which produced new state-of-the-art results in many NLP tasks on existing datasets in the general domain, using NLP to improve the processing of biomedical texts to improve inputs to LBD methods is a viable area of research. This was investigated as part of this work; details are in Chapter 3 where it is applied to biomedical NER and POS-Tagging.

2.6 Link Prediction and Knowledge Discovery

Knowledge discovery is concerned with bringing latent knowledge to the fore by putting together disparate pieces of information. LBD is a form of knowledge discovery. Knowledge discovery can take many forms which are dependent on how the existing knowledge is represented and thus how new knowledge is discovered. For the purposes of this work, we are particularly interested in knowledge discovery in graphs. This is the domain of *link prediction*.

Link prediction is the task of predicting edges or links in a graph which are not present in the current version of the graph. Liben-Nowell and Kleinberg (2003) formulated the link prediction problem in social networks and most link prediction works have focused in large part on determining which links will form next in various types of social networks where links can represent friendships (Backstrom and Leskovec, 2011; Leskovec et al., 2010), collaborations and co-authorships (Al Hasan et al., 2006; Backstrom and Leskovec, 2011), citations (Benchettara et al., 2010) and online transactions (Benchettara et al., 2010) among others. Additionally, link prediction has been used on large-scale knowledgebases to add missing data and discover new facts (Nickel et al., 2016; Schlichtkrull et al., 2018).

Link prediction has already been applied in the biomedical domain for various uses. Predicting Drug Target Interactions (DTIs) is important in repositioning existing or abandoned drugs by identifying new uses for them. Wang and Zeng (2013) and Lu et al. (2017) both used link prediction on this task by providing *in silico* predictions of interactions. Wang and Zeng (2013) used Restricted Boltzmann Machines (RBMs) to predict different types of DTIs on a multi-dimensional network while Lu et al. (2017) used similarity indices to predict links in DTI networks.

The use of link prediction for LBD has already been explored. We covered most of the previous work on this in Section 2.2.5 (Kastrin et al., 2016; Katukuri et al., 2012; Sebastian et al., 2017b). Additionally, Kastrin et al. (2014) performed link prediction on Semantic MEDLINE, a large-scale relational dataset of biomedical concepts. They used three different similarity measures as predictors for link prediction: Common Neighbours (CN), Jaccard Index/Coefficient (JI/C) and Preferential Attachment (PA). Their results showed prediction

performance which suggested plausibility of using link prediction for LBD across all their approaches, however they used the AUC (ROC) metric whose shortcomings are mentioned in Section 4.6.1. These shortcomings manifested themselves in our investigations as well in Chapter 4 where more details about link prediction will be presented.

2.6.1 Knowledge Graphs and Networks

We are concerned with link prediction in graphs because much biological data already exist as graphs and many of those which are not can be formulated as such. One such group of data are knowledgebases. These are a collection of concepts and specified relationships between them. Popular examples used in general domain text processing include DBPedia (Auer et al., 2007), WordNet (Miller, 1995), VerbNet (Schuler, 2005) etc. They have been integrated into various tasks in NLP where they provide external, usually manually-created knowledge about the real world which have boosted the performance of NLP models.

A non-exhaustive list of biomedical knowledgebases which are available for use and possibly beneficial to LBD for task such as Normalization and Entity Linking include: for Genes/Proteins, Entrez Gene (Maglott et al., 2005) and The Universal Protein Resource (Uniprot) (Wu et al., 2006); for Chemicals, Chemical Entities of Biological Interest (ChEBI) (de Matos et al., 2010; Degtyarenko et al., 2008); for Subcellular structures, the Cellular Component subontology of the Gene Ontology (Ashburner et al., 2000; Consortium et al., 2004); for Cells, The Cell ontology (Bard et al., 2005) and neXtProt's Cellosaurus (Gaudet et al., 2015); for Tissues and Anatomical structures, Foundational Model of Anatomy (Rosse and Mejino Jr, 2008; Rosse et al., 2003) and Uberon (Mungall et al., 2012); for Organisms, NCBI taxonomy (Federhen, 2012) and for Biological processes, the Biological Process subontology of the Gene Ontology (Ashburner et al., 2000; Consortium et al., 2004).

Medical Subject Headings (MeSH) have also been widely used for graph-centric LBD. SemMedDB (Kilicoglu et al., 2012), a repository of semantic predications (subject–predicate–object triples) extracted from the entire set of PubMed citations has also been used for graph centred LBD. This is unsurprising as it was billed from inception as 'a knowledge resource that can assist in hypothesis generation and literature-based discovery in biomedicine'.

2.7 Machine Learning, Neural Networks and Deep Learning

The field of Machine Learning (ML) is concerned with having machines leverage patterns in data to successfully complete a given task. Conventional machine learning techniques

were limited in their ability to process data in their raw form. Constructing machine learning systems required domain expertise as it entailed designing and engineering feature extractors that used the raw data feature vectors which were used by learning systems, like classifiers, to detect or classify patterns in the input. Representation learning methods, like neural networks, allow machines to take the raw data as input directly and automatically uncover the representations needed for detection or classification. Deep learning methods are representation learning methods with multiple levels of representation, obtained by stacking non-linear modules that transform the representation at each level (from the raw input) into a representation at a more abstract level. By composing enough of these modules, more complex functions between the input and the output can be learned. LeCun et al. (2015) and Schmidhuber (2015) provide recent overviews of deep learning in neural networks.

The most common form of machine learning approach is *supervised learning*. It works by using large collections or datasets of the items to use as input along with the respective labels. During training, the machine is shown an instance from the dataset and produces a vector of scores as output, where each component of the vector represents a score for each label category. The objective is to have the right category score the highest of all categories. To achieve this, an *objective function* that measures the error (or distance) between the output scores and the desired scores is computed. The model then modifies its internal adjustable parameters (called *weights*) to reduce this error. Central to this step is the *backpropagation* algorithm (commonly referred to as backprop) which computes the gradients of the objective function with respect to the weights. This gradient is then used to determine how each weight should be adjusted to hopefully improve performance. In a typical deep learning system, there may be hundreds of millions of weights, and hundreds of millions of labelled examples with which to train the machine learning model, although shallower neural networks would have a lot less.

Many deep learning approaches use feedforward neural network architectures, which learn to map a fixed-size input to a fixed-size output. To go from one layer to the next, a set of units compute a weighted sum of their inputs from the previous layer and pass the result through a non-linear function (in recent years the Rectified Linear Unit, ReLU (Nair and Hinton, 2010) has been popular). Units that are not in the input or output layer are called hidden units and form hidden layers. The hidden layers can be thought of as transforming the input such that categories become linearly separable by the last layer of the model.

2.7.1 Popular Neural Network Models and Topics

Although the entire fields of neural networks and deep learning have taken off as machine learning approaches of choice in recent years, some topics, particularly models, have been at the core and are relevant to understanding the work presented in this thesis.

Convolutional Neural Networks (CNNs or ConvNets)

CNNs are designed to process data that can be deconstructed as multiple channels, for example pixels in a colour image can be presented as having red, blue and green channels. Nonetheless CNNs have found uses on data with only a single channel. The architecture of a typical CNN is structured as a series of stages. The first few stages are composed of two types of layers: convolutional layers and pooling layers. Units in a convolutional layer are organized in feature maps, within which each unit is connected to local patches in the feature maps of the previous layer through a set of weights called a *filter*. The result of this local weighted sum is then passed through a nonlinearity such as a ReLU. All units in a feature map share the same filter. Different feature maps in a layer use different filters. Mathematically, the filtering operation performed by a feature map is a discrete convolution, hence the name.

Recurrent Neural Networks (RNNs) and Long Short Term Memory (LSTMs)

For tasks that involve sequential inputs, such as speech and language, it is often better to use RNNs. RNNs process an input sequence one element at a time, while their hidden units keep a *state vector* that implicitly contains information about the history of all the past elements of the sequence. RNNs are powerful models, but training them proved to be problematic because the backpropagated gradients either grow or shrink at each time step, so over many time steps they typically explode or vanish (Bengio et al., 1994) resulting in the network not learning what the training intends it to learn.

If each input element which the RNN processes is viewed as a separate unit in a neural network, it is said to be *unrolled in time*. When an RNN is unrolled in time, it can be seen as a very deep feedforward network in which all the layers share the same weights. Although their main purpose is to learn long-term dependencies, theoretical and empirical evidence shows that it is difficult to learn to store information for very long (Bengio et al., 1994). To solve this problem, one idea is to explicitly add memory to the network. The first proposal of this kind is the *LSTM* networks that uses special hidden units, which enables the network to remember inputs for a long time (Hochreiter and Schmidhuber, 1997). A special unit called the memory cell acts like an accumulator or a gated neuron: it has a connection to itself at

the next time step that allows it to copy its state and accumulates the external signal, but this self-connection is itself gated by another unit that learns to decide when to clear the content of the memory.

Distributed representations

In a neural language model, the hidden layers of the network learn to convert the input word vectors into an output word vector for the predicted next word, which can be used to predict the probability for any word in a given vocabulary to appear as the next word. The network learns word vectors that contain many active components, each of which can be interpreted as a separate semantic feature of the word, as was first demonstrated in the context of learning distributed representations for symbols (Rumelhart et al., 1986). These semantic features were not explicitly present in the input but were discovered by the learning procedure as a good way of representing the structured relationships between the input and output symbols. Learning word vectors turned out to also work very well when the word sequences come from a large corpus of real text (Bengio et al., 2003). When trained to predict the next word in sentences from a given text, the learned word vectors of semantically similar terms (such as chair and couch) are very similar. Such representations are called *distributed representations* because their features are not mutually exclusive thus allowing the distributing of the information learnt over several vectors. These word vectors are composed of learned features that were not determined ahead of time by domain specialists, but automatically discovered by the neural network. Vector representations of words learned from text are now indispensable in state-of-the-art NLP (Cho et al., 2014; Collobert et al., 2011; Sutskever et al., 2014). Work on this for biomedical NLP is detailed in Section 3.3.

2.7.2 Multi-task Learning (MTL)

One particular use of neural networks that is of interest is Multi-task Learning (MTL) (Caruana, 1993). MTL is concerned with training a ML model for multiple tasks such that the model's performance is improved either for a main task or across the multiple tasks it is trained for. This improvement can happen for multiple reasons and can be motivated in various ways. An overview of early work pertaining to MTL in neural networks was presented in (Caruana, 1997). This work motivated and laid the foundation for much of the work done in MTL by demonstrating feasibility and important early findings. Ruder (2017), which provides a more recent overview of MTL in deep neural networks, states that decades after Caruana's ground-breaking work, most MTL still follows the *hard parameter sharing*

approach proposed in (Caruana, 1993). Some of the recent uses of MTL, especially in neural networks are summarised here.

Ando and Zhang (2005) investigated learning functions which serve as good predictors of good classifiers on hypothesis spaces using multi-task learning of labelled and unlabelled data. The algorithms presented reported good results when tested on several machine learning tasks: NER, Chunking and POS-Tagging. Previous works required training sets to contain the same pattern with different labels, but this method circumvented that restriction.

According to Evgeniou et al. (2005), many empirical works, such as in (Ando and Zhang, 2005; Bakker and Heskes, 2003; Caruana, 1997), demonstrated that it is beneficial to learn multiple related learning tasks simultaneously rather than independently. They however stated that there was still much unknown about the theory behind MTL and the development of MTL methods and sought to fill this gap. In later work (Argyriou et al., 2007) they learnt multi-task features as low-dimensional representations which can be shared across a set of related tasks.

Collobert et al. (2011) built on earlier work (Collobert and Weston, 2008) and sought to use MTL in a unified model to gain increased performance in several core NLP task: NER, Chunking, POS-Tagging and Semantic Role Labelling (SRL), with neural networks. They found that while they were able to achieve a unified model which could perform all tasks without significant degradation of performance in any of them, there was little or no benefit from MTL for those tasks.

More recent work on MTL include (Maurer et al., 2016) which presented a general method for learning data representations from multiple tasks and justified their method in both multi-task learning and learning-to-learn situations. Liu et al. (2015) used multi-task deep neural networks with shared and private layers for information retrieval and semantic classification.

In the field of image processing, Zeng and Ji (2015) successfully used the weights of convolutional networks from Simonyan and Zisserman (2015) trained on general domain images as the starting point for further training on images in the biomedical domain. They reported improvements from this approach, indicating that convolutional approaches can work in multi-task settings.

The idea of MTL in neural networks has been investigated for over two decades with varying degrees of intensity and a wide spectrum of approaches. While great results have been reported in Image Processing, the results have been more muted in NLP. MTL is prospectively useful for LBD because it can be used in a variety of ways. It can be useful in the concept and relationship procuring phase as several related NLP tasks (such as POS-Tagging and Chunking) can aid in NER and even the tasks of relationship extraction and NER may be

mutually beneficial to each other. In the discovery phase, learning representations for the nodes in the graph and predicting links between them could be mutually beneficial tasks. We use MTL for the concept procuring phase by using a wide selection of biomedical datasets as different tasks to improve performance on biomedical NER (Section 3.4).

2.8 Conclusion

In summary, there has been an abundance of work on LBD since it was proposed as a strategy for discovering new scientific knowledge. These works have proposed several different approaches to LBD even when they used some similar components (Section 2.2). The evaluation methods used have been a lot less varied: there is heavy use of seeking to replicate specific discoveries especially Swanson's although there have been other approaches to evaluation as well (Section 2.3). In general, very few evaluations can lay claim to demonstrating that an approach is generalisable and practically useful. There have also been several approaches to improving LBD either through improving how it extracts useful information from text for use in LBD or in performing LBD more efficiently and obtaining high-quality proposals from the LBD process itself (Section 2.4).

The existing methods reported in the chapter either use very shallow processes to extract the entities in text either by matching to a list of known terms from lists created by external resources such as MeSH and UMLS. That makes these methods reliant on these resources being complete and updated; and makes them error-prone as simple matching will tend to produce both false positives and false negatives. Text Mining, especially NLP (Section 2.5) presents a solution to these problems and while there will be a trade-off in precision, the benefits would outweigh this downside. Link prediction (Section 2.6) is exciting because it provides both an alternative approach to facilitating simple and multi-hop LBD and a more powerful approach to knowledge discovery from biomedical knowledge represented as graphs than the traditional open and closed ABC paradigm of LBD. Advances in neural networks and deep learning (Section 2.7) have made applying all these techniques to improving LBD not only possible, but feasible and highly promising given their stellar performances on other tasks. That has been the focus of the work reported in this thesis.

While the previously stated classification by Sebastian et al. (2017a) is useful, it does not have to be rigid. Since neural networks are ubiquitous and versatile, this work sought to use them in various ways to improve various aspects of LBD. Thus some parts of this work may be considered as using the traditional approach of open and closed discovery but others less so. Chapter 4 focuses on link prediction for LBD and more powerful knowledge discovery from literature. Chapter 5 epitomises this point as it uses link prediction-inspired machine

learning models within the ABC paradigm to perform neural LBD. Additionally, all aspects sought to use neural networks in both the text processing to procure inputs phase and in the graph processing to generate discoveries phase.

This Chapter gave an overview of the relevant literature which motivated the work presented in this thesis as well as the background needed to understand the work presented. It gave details of LBD, including how it works and why it is a relevant research area. It looked at some of the previous approaches taken and proposed to improve LBD. It gave details on neural networks and deep learning whose influence and use pervades the work. It also introduced specific approaches used such as NLP, NER, link prediction and MTL. Related relevant topics like TM, knowledge discovery, representation learning, knowledgebases and graphs were also included to further orient the work in the existing literature.

Section 2.5 pointed out that LBD is a secondary process in the sense that it uses the outputs of other processes as its inputs and that the main inputs required are concepts (or entities) and relationships between those concepts. It highlighted NLP as a possible solution to some of the problems which plague concept procurement in existing LBD approaches. Chapter 3 contains the work done to improve recognising biomedical entities in unstructured text.

Chapter 3

Improving Biomedical Named Entity Recognition (NER)

3.1 Role of NER in LBD and Knowledge Discovery from Text

Concepts are a central component of LBD as they represent the objects about which knowledge is stated and discovered. These can be simple or complex and include genes/proteins, chemicals, species and diseases among others. Concepts are usually represented in text as *named entities*. Identifying which portions of unstructured text are relevant named entities and which are not is thus the necessary first step to using unstructured or weakly structured text for many computational language processing tasks, including LBD. This task is termed Named Entity Recognition (NER).

Thus far in LBD, recognizing named entities in text has revolved around using only dictionaries and ontologies such as MeSH and matching their contents to corresponding terms in text. These resources have the advantage of being vetted by humans and thus methods which utilize them can boast high precision. Unfortunately, they must be maintained and updated and this is a resource-intensive and arduous task. It is possible then for them to be inaccurate or outdated. There are also the inherent dangers of simply matching occurrences of strings of named entities in an ontology to its equivalent in text which can give rise to spurious hits. Methods incorporating NLP would almost certainly suffer lower precision but can have much better adaptability to new terms, thus increasing recall. These advantages of higher recall and no dependencies on external resources can offset the issue of lower precision. Our work in this direction sought to build on the progress of these and other machine learning techniques and models.

3.2 Neural Networks and Deep Learning for NER

There have recently been state of the art results reported on this task (Chiu and Nichols, 2016; Lample et al., 2016). Recent successful systems used word embeddings for rich input, avoided or minimized the amount of hand-crafted features used, and utilized character based models which are capable of exploiting information found at the character-level to improve performance. Our work sought to leverage these characteristics as well, along with others.

3.3 Better Word Representations for Neural Biomedical NER Models

Better inputs to neural network models can lead to improved results and has been a contributing factor to recent state-of-the-art results in NER. I played a part in an effort to improve such inputs for neural networks for various Biomedical tasks such as NER (Chiu et al., 2016a). Our role was in the extrinsic evaluation of the inputs so in the details which follow, we will focus exclusively on that aspect of the work.

As one of the main inputs of many NLP tasks, including NER, word representations have long been a major focus of research. The most recent successes have come from embedding words into a low-dimensional space using neural networks (Bengio et al., 2003; Collobert and Weston, 2008; Mikolov et al., 2013b; Pennington et al., 2014; Turian et al., 2010). These approaches represent each word as a dense vector of real numbers, which collectively form a vector space. In this vector space, words that are semantically related to each other occupy the same regions of the vector space. Among neural embedding approaches, the skip-gram model of (Mikolov et al., 2013a) has achieved cutting-edge results in many NLP tasks, such as NER, sentence completion, analogy and sentiment analysis (Fernández et al., 2014; Mikolov et al., 2013a,b).

Word embeddings have been extensively studied in recent work (e.g. (Lapesa and Evert, 2014)), but most such studies only involve general domain texts and evaluation datasets for training and evaluation. As such their results do not necessarily apply to biomedical NLP tasks. In the biomedical domain, Stenetorp et al. (2012) studied the effect of corpus size and domain on various word clustering and embedding methods, and Muneeb et al. (2015) compared word2vec and Global Vectors (GloVe) (Pennington et al., 2014), two state-of-the-art word embedding creation algorithms, on a word-similarity task. They showed that skip-gram significantly out-performs other models and that its performance can be further improved by using higher dimensional vectors. The word2vec tool was also used by Pyysalo

et al. (2013) and Kosmopoulos et al. (2015) to create biomedical domain word representations to perform NER on 3 corpora and dimensionality reduction for semantic indexing on the BioASQ datasets respectively.

Since word2vec has been shown to achieve state-of-the-art performance that can be further improved with parameter tuning, we focused on its performance on biomedical data with different inputs and hyper-parameters. We use all available biomedical scientific literature for learning word embeddings using models implemented in word2vec. Embeddings can be evaluated intrinsically and extrinsically. Intrinsic evaluation focuses on the characteristics of the vector space of the embeddings, the standard UMNSRS-Rel and UMNSRS-Sim datasets (Pakhomov et al., 2010), were used as they enabled similarity and relatedness to be measured separately. For extrinsic evaluation, a CNN NER model we developed was applied to two standard benchmark biomedical NER datasets. When the embeddings were used in biomedical NER they led to improved performance over the existing embeddings which were created with text from PMC, PubMed and Wikipedia (Pyysalo et al., 2013). The observation that a larger corpus does not necessarily guarantee better results was also made.

We used two corpora to create word vectors: the PubMed Central Open Access subset (PMC) and PubMed. PMC is a digital archive of biomedical and life science literature, which contains more than 1 million full-text Open Access articles. The PubMed database has more than 25 million citations that cover the titles and abstracts of biomedical scientific publications.

Given that the ultimate evaluation for word vectors is their performance in downstream tasks, we assessed the quality of the vectors by performing NER using two well-established biomedical reference standard datasets: the BioCreative II Gene Mention task corpus (BC2) (Smith et al., 2008) and the JNLPBA corpus (PBA) (Kim et al., 2004). Both corpora consist of approximately 20,000 sentences from PubMed abstracts manually annotated for mentions of biomedical named entities. Following the window approach architecture with word-level likelihood proposed by (Collobert and Weston, 2008), we apply a tagger built on a CNN [1]. More details of this model can be found in Section 3.4.3.

Our word vectors are used as the embedding layer of the network, with the only other input being a binary vector of word surface features. To emphasize the effect of the input word vectors on performance, we avoided fine-tuning the word vectors during training as well as introducing any external resources such as entity name dictionaries. While this causes the performance of the method to fall notably below the state of the art, we believe this minimal approach to be an effective way to focus on the quality of the word embeddings

[1] In the paper, we presented a simpler model: a feed-forward neural network, window of five words, one 300-neuron hidden layer with sigmoid activation, leading to a Softmax output. This was done to de-emphasise the role of the model in the performance. Performance dropped overall, but the findings remain the same.

created by word2vec. For parameter selection, we estimated the performance of the word vectors on the development sets of the two corpora using mention-level F-score.

We found contradictory results from changing the size of the *context window* parameter. All sets of vectors showed a notable increase in the intrinsic evaluation scores when the context window size grows. However, the extrinsic evaluation shows the opposite pattern: all results in extrinsic tasks have an early performance peak with a narrow window (the best results were *win* = 1), followed by a gradual decrease when window size increases. One possible explanation may be that a larger window emphasizes the learning of domain/topic similarity between words, while a narrow context window leads the representation to primarily capture word function (Turney, 2012). It is possible that for intrinsic evaluation datasets such as UMNSRS it is more important to model topical rather than functional similarity. Conversely, it is intuitively clear that for tasks such as NER the modelling of functional similarity such as co-hyponymy is centrally important. Further discussion on the effect of the context window size parameter can be found in (Hill et al., 2015) and (Levy et al., 2015).

In this work, we showed how the performance of word vectors changes with different corpora, preprocessing options (normal text, sentence-shuffled text, lower-cased text), model architectures (skip-gram vs. CBOW) and hyper-parameter settings (negative sampling, sub sample rate, min-count, learning rate, vector dimension, context window size). For hyper-parameter settings, it is evident that performance can be notably improved over the default parameters, but the effects of the different hyper-parameters on performance are mixed and sometimes counter-intuitive specifically in the case of NER, which is of interest to our work. It is noteworthy that (Chiu et al., 2016b) also found a similar result in general domain work with Wikipedia text.

In the next section we will focus on neural models which use the biomedical word embeddings developed from this section as inputs to perform biomedical NER. The same model used to perform extrinsic evaluation here will be used along with two expanded versions (Crichton et al., 2017).

3.4 Using Multi-task Learning to Improve Biomedical NER

High accuracy NER systems require manually annotated named entity datasets for training and evaluation. Many such datasets have been created and made publicly available. These include annotations for a variety of named entities such as genes and proteins (Smith et al., 2008), chemicals (Krallinger et al., 2015) and species (Gerner et al., 2010) names. Because manual annotations are expensive to develop, datasets are limited in size and not available for many sub-domains of biomedicine (Doğan et al., 2014; Wei et al., 2015). Consequently,

many NER systems suffer from poor performance (Batista-Navarro et al., 2015; Munkhdalai et al., 2015).

The question of how to improve the performance of NER, especially in the very common situation where only limited annotations are available, is still an open area of research. One potentially promising solution is to use multiple annotated datasets together to train a model for improved performance on a single dataset. This can help since datasets may contain complementary information that can help to solve individual tasks more accurately when trained jointly.

In machine learning, this approach is called *Multi-task Learning* (MTL) (Caruana, 1997). The basic idea of MTL is to learn a problem together with other related problems at the same time, using a shared representation. When tasks have commonality and especially when training data for them are limited, MTL can lead to better performance than a model trained on only a single dataset, allowing the learner to capitalise on the commonality among the tasks. This has been previously demonstrated in several learning scenarios in bioinformatics and in several other application areas of machine learning (Ando and Zhang, 2005; Maurer et al., 2016; Wu et al., 2015).

A variety of different methods have been used for MTL, including neural networks, joint inference, and learning low dimensional features that can be transferred to different tasks (Ando and Zhang, 2005; Argyriou et al., 2007; Evgeniou et al., 2005). Recently, there have been exciting results using CNNs for MTL and transfer learning in image processing (Zeng and Ji, 2015) and NLP (Collobert and Weston, 2008; Collobert et al., 2011; Søgaard and Goldberg, 2016), among other areas.

In this work, we investigated whether an MTL modelling framework implemented with CNNs can be applied to biomedical NER to benefit this key task. This is, to the best of our knowledge, the first application of this MTL framework to the task. Like other language processing tasks in biomedicine, NER is made challenging by the nature of biomedical texts, e.g. heavy use of terminology, complex co-referential links, and complex mapping from syntax to semantics. Additionally, the annotated datasets available vary greatly in the nature of named entities (e.g. species vs. disease), the granularity of annotation, as well as in the specific domains they focus on (e.g. chemistry vs. anatomy). It is therefore an open question whether this task can benefit from MTL.

Due to the aforementioned disparities between datasets, we treat each dataset as a separate task even when the annotators sought to annotate the same named entities. Thus the words 'datasets' and 'tasks' are used interchangeably. We first developed a single-task CNN model for NER and then two multi-task CNN models. These were applied to 15 datasets containing multiple named entities including Anatomy, Chemical, Disease, Gene/Protein and Species.

The results were then compared for evidence of benefits from MTL. On one MTL model we obtained an average F-score improvement of 0.8% with a range of -2.4% to 6.3% on MTL in comparison with single task learning from an average baseline F-score of 78.4% with range 68.6% to 83.9%. Although there is a significant drop in performance on one dataset, performance improves significantly for five datasets. For the other MTL model we obtained an average F-score improvement of 0.4% with a range of -0.2% to 1.1% on MTL in comparison with single-task learning from the same baseline. There is no significant drop in performance on any dataset, and performance improves significantly for six datasets. These are promising results which show the potential of MTL to improve biomedical NER.

3.4.1 Motivation and Background

Previous work have demonstrated the benefits of MTL. These include leveraging the information contained in the training signals of related tasks during training to perform better at a given task, combining data across tasks when few data are available per task and discovering relatedness among data previously thought to be unrelated (Bakker and Heskes, 2003; Collobert and Weston, 2008; Maurer et al., 2016). These benefits can be seen, for example, in potentially ambiguous terms which are spelled the same and are named entities in some situations, but not in others. One example of this is noted by Preiss and Stevenson (2016): the word 'cold' can mean the disease common cold, a cold sensation or the acronym for Chronic Obstructive Lung Disease. Some training sets may contain examples of both so that a model can learn to distinguish between them, but others may only contain one type. A model trained with a dataset combination which contains both types (even if each dataset contains only one type but they are opposites) can learn to distinguish between them and perform better.

We are similarly interested in these benefits, but am additionally concerned with the following, given the particular challenges of biomedical text mining:

Making the best use of information in existing datasets. Given the level of knowledge interaction and overlap in the biomedical domain, it is conceivable that signals learned from one dataset could be helpful in learning to perform well on other datasets. For example, if a multi-task model can learn to identify chemicals from a dataset which only annotates chemicals, that information can be useful when identifying Gene/Proteins in sentences which contain interaction of a Gene/Protein with a chemical.

If a model can utilize such information it could conceivably perform better as a result of having access to this additional knowledge. Currently, when models use additional knowledge as guidance it is typically hand-crafted and passed to models during training rather than learned as part of the training process.

Efficient creation and use of datasets. The datasets used to train supervised and semi-supervised models are expensive to create. They typically contain manual annotations by highly trained domain specialists (e.g. biologists with sufficient linguistics training) often covering thousands of instances (e.g. of named entities or relations) each. If models which facilitate the transfer of knowledge between existing datasets can be developed and understood, they may be able to reduce the annotation overhead. For example, such models may be able to detect which type of annotations are really needed and which are not because the information is already included in another dataset or the knowledge requirements of tasks overlap. This can help to focus annotation efforts aimed at types not covered in any existing datasets and can aid in obtaining required annotations faster even if the resulting datasets are smaller. Caruana (1997) demonstrated that *sampling data amplification* can help small datasets in MTL where tasks are related by combining the estimates of the learned parameters to obtain better estimates than it would by estimating them from small samples which may not provide enough information for modelling complex relationships between input and predictions.

It can be tempting to think that these objectives can be met by simply combining the existing corpora into a single large corpus which can then be used to train a model. The work of Wang et al. (2009), which investigated the feasibility of this for gene/protein named entities in three datasets, showed otherwise. They found that simply using combined data resulted in performance drops of nearly 12% F-score and identified as the main cause of the drop incompatibilities in the annotations due to the fact that they were made by different groups with no explicit consensus about what should be annotated.

Thus the problem of utilizing all the knowledge in existing datasets in a single model to gain the benefits of doing so, including those highlighted here, is a challenging open problem in biomedical NLP.

3.4.2 Datasets

We used 16 biomedical corpora: 15 focused on biomedical NER and one on biomedical Part-Of-Speech (POS) tagging. POS tagging is a sequential labelling task which assigns a part-of-speech (e.g. Verb, Nouns) to each word in text. We chose datasets which were publicly available and included sufficient amounts of the most utilized named entities in bioinformatics: Anatomy, Chemical, Disease, Gene/Protein and Species. The names of the datasets and information about their corresponding named entities are listed in Table 3.1. Details of their entity counts, creation, prior use, and comparison of the original data to the versions prepared for sequential labelling can be found in Appendix A.

Dataset	Contents
AnatEM (Pyysalo and Ananiadou, 2013)	Anatomy
BC2GM (Smith et al., 2008)	Gene/Protein
BC4CHEMD (Krallinger et al., 2015)	Chemical
BC5CDR (Wei et al., 2015)	Chemical, Disease
BioNLP09 (Kim et al., 2008)	Gene/Protein
BioNLP11EPI (Pyysalo et al., 2012b)	Gene/Protein
BioNLP11ID (Pyysalo et al., 2012b)	Gene/Protein, Organism Chemical, Regulon-operon
BioNLP13CG (Pyysalo et al., 2015)	Gene/Protein, Cell, Cancer, Chemical, Organism, Multi-tissue structure, Tissue, Cellular component, Organ, Organism substance, Pathological formation, Amino acid, Immaterial anatomical entity, Organism subdivision, Anatomical system Developing anatomical structure
BioNLP13GE (Kim et al., 2013)	Gene/Protein
BioNLP13PC (Ohta et al., 2013)	Gene/Protein, Chemical, Complex, Cellular component
CRAFT (Bada et al., 2012)	SO, Gene/Protein, Taxonomy, Chemical, CL, GO-CC
Ex-PTM (Pyysalo et al., 2011)	Gene/Protein
JNLPBA (Kim et al., 2004)	Gene/Protein, DNA, Cell Type, Cell Line, RNA
Linnaeus (Gerner et al., 2010)	Species
NCBI-Disease (Doğan et al., 2014)	Disease
GENIA-PoS (Ohta et al., 2002)	POS-Tags

Table 3.1 The datasets and details of their annotations

A point of concern for the stated approach would be whether there is significant overlap between the training sentences of one dataset and the test sentences in another as this would expose the model to examples which it would be evaluated on and be a confound for any performance improvement. We found that the test sets for BC5CDR and BioNLP09 overlapped with the BC2GM train sets 0.02% and 0.37%, respectively, and that the test set for JNLPBA overlapped with 0.08% of the BioNLP09 train set. These figures were not deemed large enough to influence the validity of the experiments so no steps were taken to resolve this issue.

3.4.3 Experimental Setting

We first trained a single-task model for each of the datasets in multiple settings then trained them in several MTL settings. The results of the performance in the multi-task settings were compared to those in similar single-task settings. The multi-task settings are detailed in Section 3.4.4 and involved two multi-task models which we will introduce in this section while the others involved variations on subsets of the datasets trained jointly and variation in dataset sizes.

At each training step a fixed amount of training examples (mini-batch) from the dataset being trained was selected after shuffling the training examples. For the multi-task models this mini-batch would be randomly selected from one of the datasets being trained and the model trained with only the part of the model relevant to the selected dataset activated.

The models were trained to perform NER as a sequential tagging task where each word in a sentence is tagged with an appropriate tag. The tags used were *Single-named entity*, *Begin-named entity*, *In-named entity*, *End-named entity* and *Out* where *named entity* differed according to the type of named entities in the dataset (gene/proteins, chemicals etc.). A word is tagged *Single-named entity* if it is the only word in the named entity, while entities of two or more words begin with *Begin-named entity* and end with *End-named entity*. *In-named entity* is used for words which occur between *Begin-named entity* and *End-named entity* tags if a named entity has three or more words. *Out* is used if a word is not a part of any named entity. Each dataset contained train, development and test sections and a split into these sections was introduced if none existed. Models were trained on the train section, their hyperparameters were tuned on the development section and the final evaluations were done on the test section.

The three main models in this work are all CNNs with varying architectures, and a feed-forward MLP model was used as a baseline. The models and relevant method details are described later in this section. For reasons mentioned earlier, we treated each dataset as a separate task.

The input layer of all the models accept representations of the focus word to be classified and a context of n words before and after it to give a total of $2n + 1$ words. The representations remain unchanged during training. During pre-processing, special tokens representing sentence breaks are added. The Viterbi algorithm used for calculating binary transition probabilities as by Wang et al. (2015) is applied to the outputs of all models. An overview of this is as follows, first a binary transition matrix is calculated from the training data labels where for each possible tag transition sequence a score of 1 is given if the training data contains the transition and 0 if such a transition does not exist. The information in this matrix is then applied to the sequence of predicted tags and used to update any predicted tag

sequences which are not seen in the training data (i.e. with tag transition score 0) with a tag transition sequence which was seen.

Baseline Model

This was a Multi-layer Perceptron (MLP) network with a hidden Rectified Linear Unit (ReLU) activation layer leading to an output layer with Softmax activation.

Single-task Model

The input layer leads to a convolutional layer which applies multiple filter sizes to a window of words in the input in a single direction. To apply each filter in only a single direction over the window of words, the width of the filter always equals the amount of dimensions of the word embeddings. The outputs of all filters then go to a layer with ReLU activation. The outputs are then concatenated and reshaped before they pass into a fully connected layer then an output layer with a Softmax activation which classifies the focus word by selecting the label with the maximum value of the Softmax output. This model is similar to the one used by Collobert and Weston (2008) but there is no max-pooling after the convolution layer. We refrained from using pooling layers so that positional information in the input would not be lost. We experimented with max-pooling and found that performance improved when it was not used. See Figure 3.1 for a depiction of this model.

Multi-output Multi-Task Model

The first multi-task model is similar to the single-output model described in Section 3.4.3 up to the fully-connected layer. In this model there are separate fully-connected and output layers for each task the model learns. Thus a private output layer with Softmax activation represents each task but all tasks share the rest of the model. This model is similar to the one used by Collobert et al. (2011) but there are convolutional layers. It is also similar to the one used by Collobert and Weston (2008) but we share the convolution layers in addition to the word embeddings and there is again no max-pooling. Figure 3.2 depicts this model.

Dependent Multi-Task Model

This model makes use of the fact that some NLP tasks are able to use information from other tasks to perform better. An example of this is that NER may utilize the information contained in the output of POS tagging to improve its performance. This model combines two of the single-task models described above with one model accepting input from the other. The

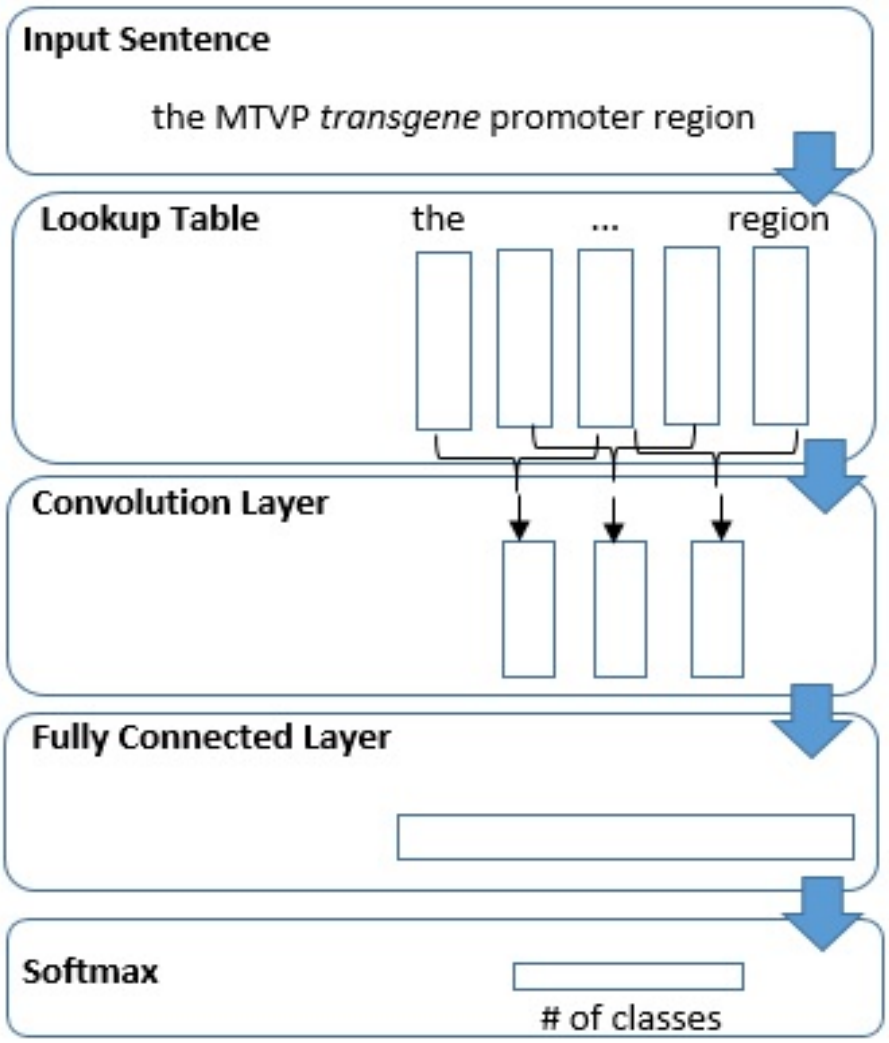


Fig. 3.1 Single-task Convolutional Model

first model trains for the auxiliary task which is POS tagging in this case, then that trained model is used in the training of the second part of the model for the main task, NER in this case. This is done by concatenating the fully connected layers of the model trained for the auxiliary task and the one trained for the main task. The use of this arrangement is similar to the one used in (Huang et al., 2013) but these layers between word embeddings and Softmax are convolutions and fully-connected layers. See Figure 3.3 for a depiction of this model.

3.4.4 Experiments

All inputs consisted of a focus word and 3 words to the left and right of it to give a 7 word context window. The baseline model had one hidden layer of size 300 and was trained with

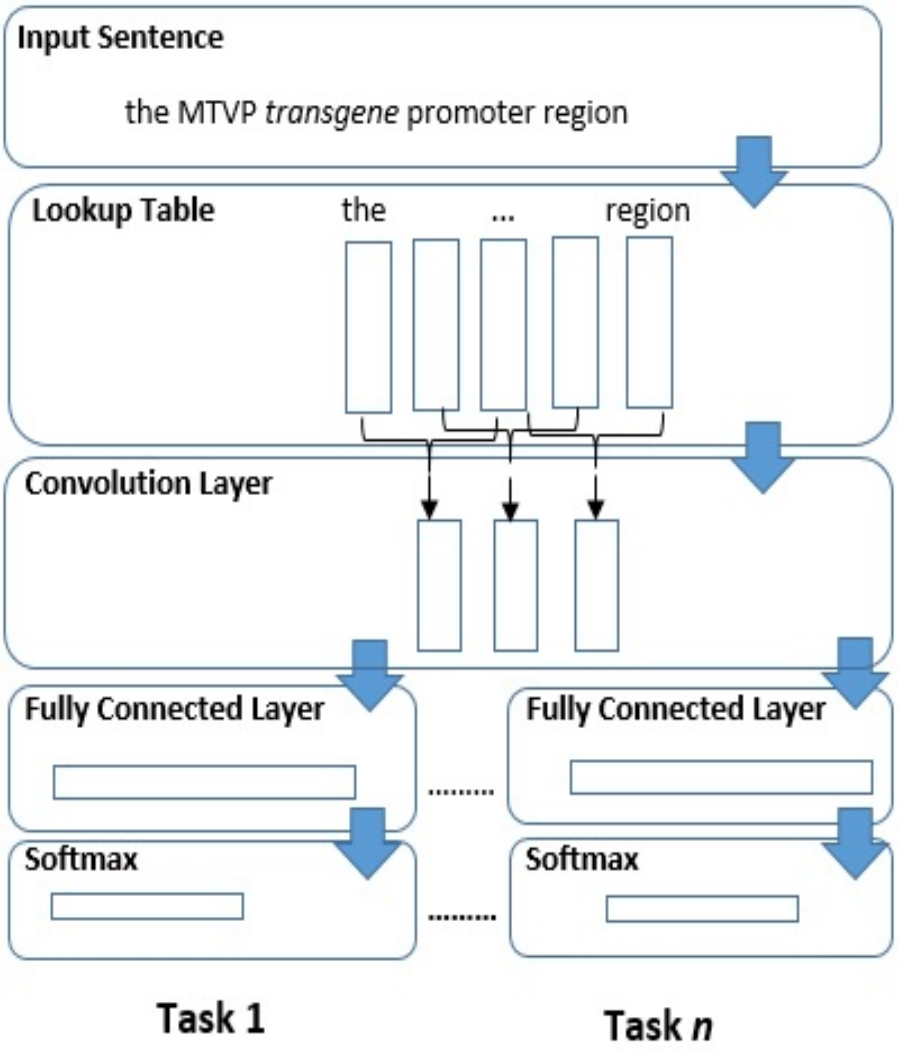


Fig. 3.2 Multi-output Multi-Task Convolutional Model

the Stochastic Gradient Descent (SGD) optimizer using mini-batch size 50. All CNN models used Dropout (Srivastava et al., 2014) with a probability of 0.75 at the fully connected layer only. No other form of regularization was used. The CNN models used 100 filters of sizes of 3, 4 and 5 and a learning rate of 10^{-4} was used with the Adam (Kingma and Ba, 2015) optimizer on mini-batch size 200. The Categorical Crossentropy loss function was used. These settings were chosen as they produced the best results from parameter tuning on the development sections of BC2GM, BioNLP09, BC5CDR and AnatEM.

Each dataset was used to train a single-task model (Section 3.4.3). Details of these as well as the various multi-task experiments utilizing multi-task models (also Section 3.4.3) follow.

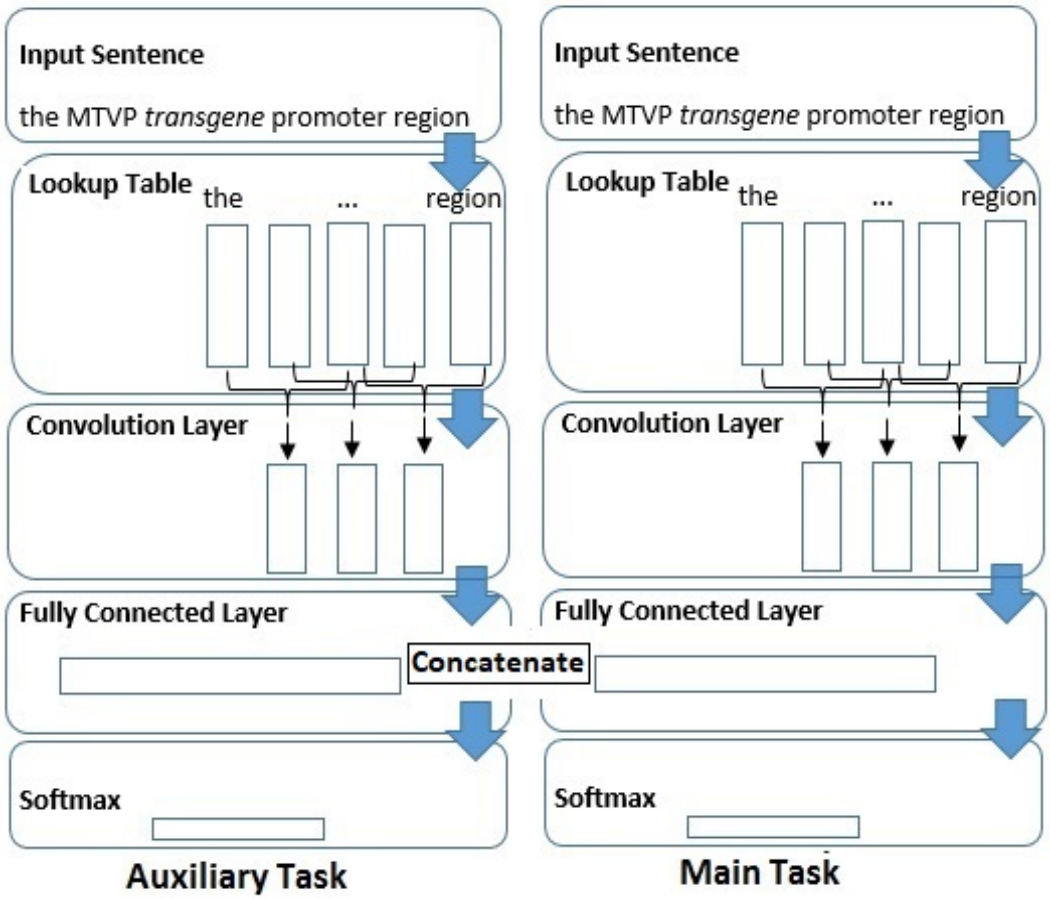


Fig. 3.3 Multi-task Dependent Convolutional Model

Baseline experiments: We completed tests with the baseline model using each of the datasets listed in Table 3.1.

Effect of datasets on each other: To determine the exact effect that each NER dataset had on every other one, the multi-output multi-task model was used to train each NER dataset with every other one. That is, a Multi-output multi-task model was trained for each ordered combination of the datasets to give 15 x 14 models.

Grouping datasets with similar named entities: Several datasets in Table 3.1 sought to annotate the same named entities (Chemicals, Cells, Cellular Component, Gene/Proteins, Species). Modified versions of these datasets which extracted only those entity annotations and then grouped the datasets which annotated the same named entity were created. This was done by changing the labels of the classes of annotations of entities, other than the one in focus, to the 'Out' class. These groups were used to train the Multi-output multi-task model from Section 3.4.3.

Multi-task experiments with complete dataset suite: The first part of this experiment used all the NER datasets to train the Multi-output multi-task model. In the second part, the Dependent multi-task model was used to train each dataset with the GENIA-PoS dataset as the auxiliary task.

Correlation of dataset size and effect of MTL: To determine how the effect of MTL varies with dataset size for the chosen datasets, we used only 50%, 25% and 10% of the training section of each dataset in both single and multi-task settings and observed the effect this had on performance. In the multi-task settings, the reduced dataset was trained only with the dataset which best improved it as determined from the effects experiment described above (i.e. the dataset listed in the 'Best Dataset' column of Table 3.2). The Multi-output multi-task model (Section 3.4.3) was used for these experiments.

3.4.5 Results and Discussion

In the tables of results, columns headed STM refer to results from the single-task model, columns headed MO-MTM refer to results from the Multi-output multi-task model and columns headed D-MTM refer to the Dependent multi-task model. The scores reported are macro F1-Scores (a single precision and recall calculated for all types) of the entities at the mention level so exact matches are required for multi-word entities. Best results are shown in **bold** and statistically significant score changes are shown with an asterisk (*). All statistical tests were done using a two-tailed t-test with $\alpha = 0.05$. The accuracy on the POS tagging task for the model used in the Dependent multi-task model training was 98.10%.

Multi-task Learning effect of each Dataset

Recall from Section 3.4.4 that in this experiment the multi-output multi-task model was used to train each NER dataset with each of the other 14 NER datasets in the suite to determine which of the other 14 datasets produced the best result when jointly trained with the other.

Information about the maximum scores achieved for each dataset is shown in Table 3.2. In 4 of the 15 datasets, there were maximums which were significantly higher than the single-task maximum scores shown in the 'STM' column of the table. This illustrates that for these datasets there is at least one other dataset in the suite which could be trained jointly with it which would yield better performance than training it by itself. It should be noted here that quite a large number of experiments were carried out to obtain the results in this table and while the differences in performance were significantly better, the large number of datasets increase the probability of hitting upon another dataset in the suite which can have that positive effect.

Dataset	STM	Best MO-MTM	Best Dataset
AnatEM	81.55	**81.68**	NCBI-Disease
BC2GM	**72.63**	72.21	Ex-PTM
BC4CHEMD	**82.95**	80.31	BioNLP13GE
BC5CDR	83.66	**83.77**	BioNLP11EPI
BioNLP09	83.90	**84.16**	BioNLP13GE
BioNLP11EPI	77.72	**78.10**	BioNLP09
BioNLP11ID	81.50	**82.26***	BioNLP13GE
BioNLP13CG	76.74	**77.33***	BioNLP13PC
BioNLP13GE	73.28	**76.09***	BioNLP11EPI
BioNLP13PC	80.61	**80.94**	Ex-PTM
CRAFT	**79.55**	78.48	BioNLP13GE
Ex-PTM	68.56	**73.58***	BioNLP11EPI
JNLPBA	**69.60**	68.92	BioNLP13GE
Linnaeus	**83.98**	83.63	NCBI-Disease
NCBI-Disease	80.26	**80.74**	Ex-PTM
Average	78.43	78.81	N/A

Table 3.2 Best Positive Effects. Datasets in rightmost column are the auxiliary ones. (**Bold**: best scores, *: statistically significant)

An aim of this experiment was to determine which dataset had the most positive interaction with a particular dataset. Table 3.2 shows the result of this in the 'Best Dataset' column. Most of the datasets which proved to be the best combined with a given dataset were predictable in that datasets which annotated the same named entities were able to help each other, but other successful combinations were less predictable, for example the dataset which best interacted with BC4CHEMD (Chemical) was BioNLP13GE (Gene/Protein) despite the presence of other datasets which annotated Chemicals and the dataset which best interacted with Linnaeus (Species) was NCBI-Disease (Disease) not another dataset which annotated Species.

The full list of results from the 15 x 14 models were not included here for brevity, but they can be found in Section A.2 of Appendix A.

Multi-task Learning in grouped datasets

Recall from Section 3.4.4 that in this experiment we took datasets which annotated the same named entity and trained them all jointly using the multi-output multi-task model, for example Table 3.3 refers to an experimental setup where only datasets which annotated Chemicals were jointly trained and evaluated on each of the test sets in turn.

Dataset	STM	MO-MTM
BC4CHEMD	**82.95**	82.51
BC5CDR	87.02	**89.22***
BioNLP11ID	**65.79**	63.74
BioNLP13CG	66.40	**77.17***
BioNLP13PC	74.53	**79.46***
CRAFT	**80.00**	74.83
Average	76.43	77.49

Table 3.3 Chemical Group. (**Bold**: best scores, *: statistically significant)

Dataset	STM	MO-MTM
BioNLP11ID	74.14	**77.25***
BioNLP13CG	82.75	**86.29***
CRAFT	**97.74**	97.44
Linnaeus	**83.98**	83.54
Average	84.65	86.13

Table 3.4 Species Group. (**Bold**: best scores, *: statistically significant)

The results in Tables 3.3 to 3.8 present the effect of training the Multi-output model with datasets which aim to annotate similar named entities. In four of the six groups, there were marked increases in the average performance of the group of tasks, marked decrease in one group and the results of the remaining one were equivalent. Across the groups there were 27 experiments; 16 showed significant increase, 1 showed significant decrease and the remaining 10 showed no significant change.

It is important to note that although the focus of the annotations were similar, both the sources of the text and the annotations are different for these datasets. This general improvement suggests that the multi-task model was able to utilize the real-world distributions from which these labelled examples were sampled and leverage information in all or some of them to increase performance in most of them, despite variations in source text and possibly annotation guidelines. This provides evidence of MTL having a positive effect on the NER task.

Dataset	STM	MO-MTM
BioNLP13CG	72.79	**74.80***
BioNLP13PC	83.23	**84.67***
CRAFT	61.04	**63.08***
Average	72.35	74.18

Table 3.5 Cellular Component Group. (**Bold**: best scores, *: statistically significant)

Dataset	STM	MO-MTM
BC5CDR	**80.46**	80.39
NCBI-Disease	80.26	**80.46**
Average	80.36	80.42

Table 3.6 Disease Group. (**Bold**: best scores, *: statistically significant)

Dataset	STM	MO-MTM
BioNLP13CG	**83.25**	82.83
CRAFT	**88.08**	86.89*
Average	85.66	84.86

Table 3.7 Cell Group. (**Bold**: best scores, *: statistically significant)

Dataset	STM	MO-MTM
BC2GM	72.63	**73.04**
BioNLP09	83.90	**84.76***
BioNLP11EPI	77.72	**79.00***
BioNLP11ID	86.20	**87.21***
BioNLP13CG	83.40	**85.98***
BioNLP13GE	73.28	**79.66***
BioNLP13PC	83.21	**84.84***
CRAFT	72.85	**75.16***
Ex-PTM	68.56	**74.91***
JNLPBA	69.60	**69.73**
Average	77.14	79.43

Table 3.8 Gene/Protein Group. (**Bold**: best scores, *: statistically significant)

Multi-task Learning on all datasets

Recall from Section 3.4.4 that in this experiment we trained the Multi-output multi-task model and the Dependent multi-task model with all the datasets as they were originally annotated by randomly selecting a particular dataset at each training step and training the model on the relevant parts activated. Table 3.9 show the results of this.

These results show that the average score of the Multi-output model is higher than that of the 15 separately trained models. Since the average score over such varied datasets as those used can be misleading, we examined each dataset individually and analysed the differences in performance.

This revealed that of the results for individual datasets, there were 6 where the difference in performance between the Multi-output model and the single-task model was statistically significant. There were 5 datasets where it performed significantly better and 1 dataset where it was significantly worse. The performances in the 9 remaining datasets were comparable. This also provides evidence of MTL having a positive effect on the NER task as in the previous experiment, but in this case it is a more impressive feat since the number of datasets and the variability among them are more pronounced here.

Table 3.9 also illustrates that the average score of the Dependent model was higher than that of the 15 separately trained models. Analysis of the results revealed that of the results for individual datasets, there were 6 where the difference in performance between that and the single-task model was significant. In all 6 it performed significantly better, it was significantly worse in none and the performances in the 9 remaining datasets were comparable.

These results show the advantages and disadvantages of the two approaches to MTL which each model incorporates. In the Dependent model the average improvement was less impressive than the Multi-output model but it also shows that this model did not make performance on any particular dataset significantly worse. This is possibly due to the large amount of separation between the components responsible for each task which allows for the NER model to incorporate POS information when it can be helpful and ignore it when it is not. Comparison of the results of the Multi-output model and the Dependent Model show that the Multi-output model had a higher average score because it gave larger gains in the datasets where it performed better but also showed larger losses where it did not. This is possibly due to sharing most of the model among the datasets regardless of whether or not this is helpful. This result indicates that in cases where tasks are thought to be similar and can contribute equally the Multi-output model may be the better of the two while in cases where there is a clear main and auxiliary task separation, the Dependent model may perform better.

Dataset	**Baseline**	**STM**	**MO-MTM**	**D-MTM**
AnatEM	81.79	81.55	81.83	**82.21***
BC2GM	70.31	72.63	**73.17**	72.87
BC4CHEMD	81.08	82.95	82.37	**83.02**
BC5CDR	83.11	83.66	**83.90**	83.83
BioNLP09	81.84	83.90	**84.20**	84.10
BioNLP11EPI	74.98	77.72	**78.86***	78.03*
BioNLP11ID	81.44	81.50	80.58*	**81.73**
BioNLP13CG	75.23	76.74	**78.90***	77.52*
BioNLP13GE	72.49	73.28	**78.58***	74.00*
BioNLP13PC	79.35	80.61	**81.92***	81.50*
CRAFT	78.76	79.55	79.10	**79.56**
Ex-PTM	65.75	68.56	**74.90***	69.67*
JNLPBA	67.43	69.60	**70.09**	69.44
Linnaeus	79.01	83.98	81.57	**84.04**
NCBI-Disease	79.09	80.26	79.02	**80.37**
Average	76.78	78.43	79.26	78.79

Table 3.9 Single Task and Multi-Task F-Scores on NER tasks. (**Bold**: best scores, *: statistically significant compared to single-task model)

There were seven datasets which showed significant performance change across the two multi-task models. Five of them (BioNLP11EPI, BioNLP13CG, BioNLP13GE, BioNLP13PC, Ex-PTM) were improved in both models which indicated that these datasets benefited from simply having the information present in the additional datasets available to them, regardless of the model. One (AnatEM) had better performance in the Dependent model but no difference in the Multi-output model while another (BioNLP11ID) had significantly worse performance in the Multi-output model but no significant performance change in the Dependent model. Both of these datasets recorded improved performance in the Dependent model which indicate that they benefit from having POS-Tagging information integrated in the manner which the Dependent model uses.

Dataset size and Multi-task Learning

Recall from Section 3.4.4 that in this experiment we used only 50%, 25% and 10% of the training section of each dataset in both single and multi-task settings and observed the effect this had on performance. So for example, we trained the single task model on AnatEM with only 50, 25 and 10% of its original training data and compared these results to it trained in a multi-task setting with 50, 25 and 10% of its original training data along with the full training

section of the dataset it performed best on in the first experiment (Table 3.2), NCBI-Disease in this example.

Table 3.10 correlates dataset performance and decreased size both in isolation and when trained in a multi-task setting. The best scores for each dataset is in bold and the better scores for each training set size are italicized. Statistically significant changes in scores relative to the full single-task model are shown with asterisks while statistically significant changes in scores relative to the corresponding single-task model are marked with a +.

Multi-task Learning is advantageous here as well as shown in the '0.5 MO-MTM', '0.25 MO-MTM' and '0.1 MO-MTM' columns. As the size of the datasets were reduced, the multi-task model was able to show an increase in average score over the corresponding single-task models. The gap between the average scores of the single-task models and the corresponding multi-task model also widened as the datasets became smaller. In fact, there were two datasets (BioNLP13GE and Ex-PTM) where using only 50% of the training data in a multi-task setting yielded significantly better performance than using the full training data in a single task setting. In the case of Ex-PTM, this was also the case when it was used with only 25% of its training data. This augurs well for our stated aim of using MTL to improve performance on small datasets. It can also indicate that new datasets can contain fewer annotations and thus would consume less resources to create while still being effective - another stated aim of this work.

An additional result from this experiment was that, for many of the datasets, randomly removing 50% of the training data sentences resulted in an average drop of only approximately 3.4% F-score in single task training as can be seen by comparing the '1.0 STM' and '0.5 STM' columns of Table 3.10. When the model is trained on 75% less training data, that average drop extends to 8% as some datasets continue to be robust although there is a predictable drop in performance in most datasets. It is not until 90% of the training data of the datasets are removed that a steep drop in average performance of approximately 16.7% is registered across all datasets. This high performance on reduced-sized corpora supports what is reported in (Leaman et al., 2009) using BANNER (Leaman and Gonzalez, 2008), a NER model based on Conditional Random Fields (CRF) for biomedical NER. This may indicate that, like BANNER, the single-task model presented in Section 3.4.3 is able to efficiently utilize even a relatively small amount of training data to obtain good enough performance. It is important to note that in the respective data reduction scenarios, the multi-task models record drops of approximately 0.2% when 50% of the training data is removed, approximately 3.0% when 75% is removed and approximately 9.8% when 90% is removed.

There are two caveats which temper these results. The first is that the multi-task model would have more training data at its disposal than the reduced training data of the single-

Dataset	1.0 STM	0.5 STM	0.5 MO-MTM	0.25 STM	0.25 MO-MTM	0.1 STM	0.1 MO-MTM
AnatEM	**81.55**	*78.74**	78.35*	74.82*	*76.59*+*	*65.99**	63.15
BC2GM	**72.63**	70.27*	*70.73*+*	*67.37**	67.14*	63.07*	*63.14**
BC4CHEMD	**82.95**	*80.16**	79.22*+	*76.81**	76.26*	71.94*	*72.53**
BC5CDR	**83.66**	81.15*	*82.45*+*	79.09*	*80.44*+*	74.47*	*75.48**
BioNLP09	**83.90**	81.89*	*82.22**	*80.56**	79.58*	75.12*	*78.32**
BioNLP11EPI	**77.72**	74.00*	*77.57*+*	70.89*	*75.61+*	67.63*	*75.04*+*
BioNLP11ID	**81.50**	76.65	*81.39*	70.60*	*78.17*+*	68.19*	*73.52**
BioNLP13CG	**76.74**	70.58*	*75.02*+*	65.08*	*72.98*+*	51.61*	*67.86*+*
BioNLP13GE	73.28	73.32	***81.37*+***	67.43	*78.80**	52.66*	*77.12*+*
BioNLP13PC	**80.61**	75.39*	*77.57*	70.03*	*73.90**	57.62*	*68.65*+*
CRAFT	**79.55**	75.25*	*79.01+*	72.19*	*76.79*+*	60.91*	*71.00**
Ex-PTM	68.56	62.81	***74.60*+***	53.30*	*74.27*+*	47.01*	*69.83+*
JNLPBA	69.60	68.34	***69.65***	66.63*	*68.13*	62.80*	*65.40*+*
Linnaeus	83.98	80.08*	***87.61+***	69.53*	*79.86*	39.44	*45.73*
NCBI-Disease	**80.26**	76.51	*76.84*	71.88*	*73.55**	*67.48**	62.89*
Average	78.43	75.01	78.24	70.41	75.47	61.73	68.64

Table 3.10 Effect of dataset size reduction on Single-Task and Multi-task performance. (**Bold**: best scores for dataset, *Italic:* better score for each setting, *****: statistically significant compared to full single-task model, **+**: statistically significant compared to corresponding single-task model)

task models and this situation would be exacerbated when the second dataset is quite large. Additionally, given the wide range of dataset sizes, the absolute sizes of the reduced training sets would vary from dataset to dataset. The second is that there is the potential for some datasets to be more compatible with the testing data for another dataset as they could have been drawn from the same source sentences or annotated in similar efforts.

Applications and Practicality

The argument can be made that the increases in performance we report are trivial and may not be worth doing in practical applications. This can be especially true of the Dependent multi-task model. We note however that, if there is no benefit from Multi-task Learning, then the single-task setting can be used for a particular task and the practitioner is no worse off than before. Our contribution is that for some datasets the benefits can be significant and in those cases we present an option to the practitioner to obtain improved performance which previously was not available. An additional argument against application of the work presented is the results which show that it can be difficult to predict when MTL will be beneficial and by how much. We contend that the models and methods presented here make

it possible to quickly determine empirically the amount of benefit that MTL, as implemented here, provides.

The training time of the models varied according to the size of the dataset(s) involved and the type of model. The experiment which took the longest time to run was the one where all the datasets were trained together with the Multi-output multi-task model which were ran for 190,000 steps with batch sizes of 200 examples drawn on each step from a randomly selected dataset. This took approximately 40 minutes to train on a single Nvidia Titan X GPU. As the weights are randomly initialized at the start of training, there is some variation in scores between runs. For the single task experiments, the average variance in F-Score was 0.099. For the Multi-output multi-task model it was 0.092 and for the Dependent multi-task model it was 0.012. In our experiments under the conditions outlined here, training never failed entirely.

The models were developed in Python (Van Rossum and Drake Jr, 1995) using Keras (Chollet et al., 2015) with Tensorflow (Abadi et al., 2015) backend. The Numpy (Oliphant, 2006) library was also used. The code for the models used in this work can be found at https://github.com/cambridgeltl/MTL-Bioinformatics-2016.

3.4.6 Multi-task Learning Conclusion

In this work we investigated whether Multi-task Learning could benefit the key text mining task of biomedical NER across various NER datasets. We first developed a single task CNN model for NER and then two variants of a multi-task CNN. We trained these on 15 domain-specific datasets representing a smorgasbord of biomedical named entities.

We observed an average improvement on MTL in comparison with single task learning. Individually, there were also significant improvements on many of the datasets. Although there was a drop in performance on some tasks, for most tasks performance improves significantly. This is a promising result which shows the potential of MTL for biomedical NER.

Limitations to the work include that it can be difficult to predict situations when these MTL models will definitely provide benefit and the extent of any increases in performance that they may give before it is actually applied. This area has recently received research attention (Alonso and Plank, 2017; Bingel and Søgaard, 2017; Luong et al., 2016) and some of the proposed methods may be useful in this regard in the future. Another limitation is that the current implementation of the models does not allow for overlapping annotations of the same term in the data.

3.5 Character-level Deep Learning Model for General and Biomedical NER

The results from the previous section were promising for the task of biomedical NER. However the models used there were convolutional neural networks and it is generally accepted that RNNs, especially Long Short-Term Memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) are better for sequential data such as text, hence NLP tasks like NER. Additionally, recently there has been progress in incorporating character-level features into neural networks for various NLP tasks like NER.

Since biomedical entities encode much information at the character level (e.g. names ending with 'ase' tend to be enzymes), it was a logical next step to 1) use LSTMs for biomedical NER and 2) incorporate character-level features into LSTMs. This led to a role in work which utilised attention for character-level LSTMs for several sequence labelling tasks including biomedical NER (Rei et al., 2016). The relevant parts are described in this section.

3.5.1 Attention-based Character-level Model for Biomedical NER

Many NLP tasks, including NER, POS-tagging and shallow parsing can be framed as sequence labelling so the development of accurate and efficient sequence labelling models is thus useful for a wide range of downstream applications. Work in this area has traditionally involved task-specific feature engineering such as integrating gazetteers for NER, or using features from a morphological analyser in POS-tagging. Recent developments in neural architectures and representation learning have led to the proliferation of models that can discover useful features automatically from data. Such sequence labelling systems are applicable to many tasks, using only the surface text as input, yet are able to achieve competitive results (Collobert et al., 2011; Irsoy and Cardie, 2014).

Current neural models make use of word embeddings as explained in Section 3.3, which allow them to learn similar representations for semantically or functionally similar words. While this is an important improvement over count-based models, they still have weaknesses that should be addressed. The most obvious problem arises when dealing with a previously unseen token, referred to as *out-of-vocabulary (OOV)* words. In such cases the model does not have an embedding and needs to back-off to the same, generic representation for all OOV words. Words that have been seen very infrequently have embeddings, but they will likely have low quality due to lack of training data. The approach can also be sub-optimal in terms of parameter usage – for example, certain suffixes indicate more likely POS tags

for these words, but this information is repeatedly encoded into each embedding instead of being shared between the whole vocabulary.

In this work, we constructed a task-independent neural network architecture for sequence labelling, and then extend it with two different approaches for integrating character-level information. By operating on individual characters, the model is able to infer representations for previously unseen words and share information about morpheme-level features. This can be particularly useful for handling unseen words – for example, if it have never seen the word cabinets before, a character-level model could still infer a representation for this word if it has previously seen the word cabinet and other words with the suffix -s. In contrast, a word-level model can only represent this word with a generic out-of-vocabulary representation, which is shared between all other unseen words. We proposed a novel architecture for combining character-level representations with word embeddings using a gating mechanism, also referred to as *attention*, which allows the model to dynamically decide which source of information to use for each word. Additionally, it harnesses a new objective for model training where the character-level representations are optimised to mimic the current state of word embeddings so that the initially learnt information is not lost.

The neural models were evaluated on some of the biomedical NER datasets and the POS-tagging dataset described in Table 3.1, among others. Our experiments show that including a character-based component in the sequence labelling model provides substantial performance improvements on all the benchmarks.

3.5.2 Model

The basic model is a word-level neural network for sequence labelling, following the models described in (Lample et al., 2016; Rei and Yannakoudakis, 2016). It receives a sequence of tokens as input, and predicts a label corresponding to each of the input tokens. The tokens are first mapped to a distributed vector space, resulting in a sequence of word embeddings. Next, the embeddings are given as input to a bi-directional LSTM (two LSTM components moving in opposite directions through the text), which create context-specific representations. The respective forward- and backward-conditioned representations are concatenated for each word position, resulting in representations that are conditioned on the whole sequence. Following Huang et al. (2015), we also used a CRF as the output layer, which conditions each prediction on the previously predicted label. The last hidden layer is used to predict confidence scores for the word having each of the possible labels. A separate weight matrix is used to learn transition probabilities between different labels and the Viterbi algorithm, as described in Section 3.4.3 and used by the CNN models, is used to find an optimal sequence of weights.

Distributed embeddings still treat words as atomic units and ignore any surface- or morphological similarities between different words but by constructing models that operate over individual characters in each word, one can take advantage of these regularities. At the time this work was done, research into character-level models was still in fairly early stages, and models that operate exclusively on characters were not yet competitive to word-level models on most tasks so, instead of fully replacing word embeddings, combining the two approaches made sense. This allowed the model to take advantage of information at both levels of granularity. Each word is broken down into individual characters, these are then mapped to a sequence of character embeddings, which are passed through the bidirectional LSTM.

This approach assumes that the word-level and character-level components learn somewhat disjoint information, and it is beneficial to give them separately as input to the sequence labeller Alternatively, we can have the word embedding and the character-level component learn the same semantic features for each word. Instead of concatenating them as alternative feature sets, the network was specifically constructed so that they would learn the same representations, and then allow the model to decide how to combine the information for each specific word. Instead of concatenating the character embedding with the word embedding, the two vectors are added together using a weighted sum, where the weights are predicted by a two-layer network. The main benefits of character-level modelling are expected to come from cases where useful information is encoded at the character level and improved handling of rare and unseen words, whereas frequent words are likely able to learn high-quality word-level embeddings directly. To take advantage of this, and train the character component to predict these word embeddings the attention-based architecture requires the learned features in both representations to align, and an extra constraint is added to encourage this. During training, a term is added to the loss function that optimises the character vector to be similar to the word embedding. Importantly, this is done only for words that are not out-of-vocabulary since it is desirable for the character-level component to learn from the word embeddings, but this should exclude the OOV embedding, as it is shared between many words. The basic model and its two variants are depicted in Figure 3.4.

3.5.3 Results

Optimising the hyperparameters for each dataset separately would likely improve individual performance, but we conducted more controlled experiments on a task-independent model so we use the same hyperparameters on all datasets, and the development set is only used for the stopping condition. With these experiments, the aim was to determine 1) which sequence labelling tasks character-based models offer an advantage to, and 2) which character-based

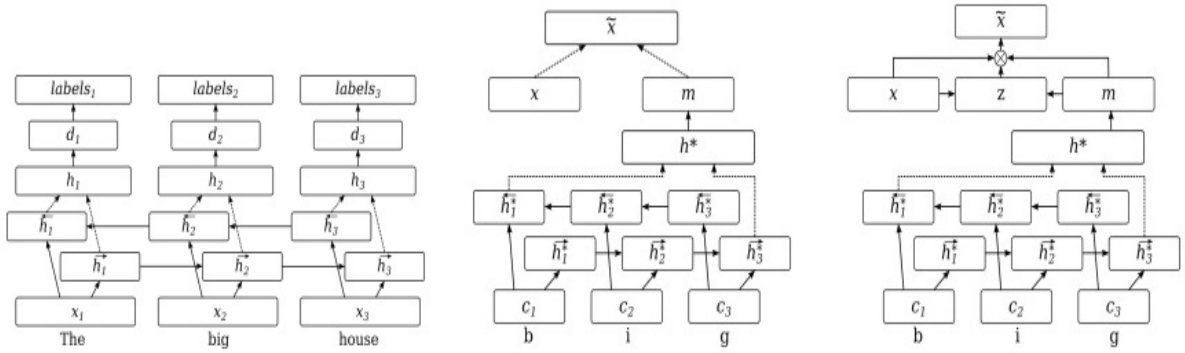


Fig. 3.4 Left: Basic neural sequence labelling model. Middle: concatenation-based character architecture. Right: attention-based character architecture. Dotted lines show vector concatenation. *x* represent word vectors.

architecture performs better. Results for the different model architectures on the biomedical NER and POS-tagging datasets are shown in Table 3.11. As can be seen, including a character-based component in the sequence labelling architecture improves performance on every benchmark. The NER datasets have the largest absolute improvement; we believe that this is due to several factors. The model is able to learn character-level patterns to deduce names of biomedical entities it did not see in training as they have the general form of biomedical entity names, and also improve the handling of any previously unseen tokens. Compared to concatenating the word- and character-level representations, the attention-based character model outperforms the former on all evaluations. The mechanism for dynamically deciding how much character-level information to use allows the model to better handle individual word representations, giving it an advantage in the experiments. Visualisation of the attention values shows that the model is actively using character-based features, and the attention areas vary between different words. The results of this general tagging architecture are competitive, even when compared to previous work using hand-crafted features. The network achieves 72.70% on JNLPBA compared to 72.55% in (Zhou and Su, 2004). In the case of BC2GM, we were also able to beat the previous best results – 87.99% compared to 87.48% in (Campos et al., 2015).

3.5.4 Attention-based Character-level Approach Conclusion

Developments in neural network research allow for model architectures that work well on a wide range of sequence labelling datasets, such as NER, without requiring hand-crafted data. While word-level representation learning is a powerful tool for automatically discovering

Dataset		Word-based	Character Concatenation	Character Attention
BC2GM	**Dev**	84.07	87.54	**87.98**
	Test	84.21	87.75	**87.99**
BC4CHEMD	**Dev**	78.63	82.80	**83.75**
	Test	79.74	83.56	**84.53**
JNLPBA	**Dev**	75.46	76.82	**77.38**
	Test	70.75	72.24	**72.70**
GENIA-POS	**Dev**	97.55	98.59	**98.67**
	Test	97.39	98.49	**98.60**

Table 3.11 Comparison of word-based and character-based sequence labelling architectures on 4 biomedical sequence labelling datasets. (**Bold**: best scores for dataset)

useful features, these models still come with certain weaknesses – rare words have low-quality representations, previously unseen words cannot be modelled at all, and morpheme-level information is not shared with the whole vocabulary.

In this work, we investigated character-level model components for a sequence labelling architecture, which allow the system to learn useful patterns from sub-word units. In addition to a bidirectional LSTM operating over words, a separate bidirectional LSTM is used to construct word representations from individual characters. A novel architecture for combining the character-based representation with the word embedding by using an attention mechanism, allowing the model to dynamically choose which information to use from each information source was described. In addition, the character-level composition function is augmented with a novel training objective, optimising it to predict representations that are similar to the word embeddings in the model. The evaluation was performed on different biomedical datasets representing 2 sequence labelling tasks. It was found that incorporating character-level information into the model improved performance on every benchmark, indicating that capturing features regarding characters and morphemes is indeed useful in a general-purpose tagging system. In addition, the attention-based model for combining character representations outperformed the concatenation method used in previous work in all evaluations.

3.6 Investigations into MTL, Character-level Attention-based NER

Table 3.12 shows the comparison of the best scores obtained through multi-task learning with CNNs described in Section 3.4 and those obtained by integrating character-level attention

Dataset	Character Attention Bi-LSTM	Multi-task Learning CNN
BC2GM	**87.99**	73.17
BC4CHEMD	**84.53**	83.02
JNLPBA	**72.70**	70.09

Table 3.12 Comparison of character-based sequence labelling results with MTL on the 3 biomedical NER datasets used in both works. (**Bold**: best scores for dataset)

using bi-directional LSTMs. These are the three NER datasets that were used by both works, and the character-level attention using bi-directional LSTMs performed better in all cases. Since character-level attention using bi-directional LSTMs and MTL are not mutually exclusive, an obvious next step is to introduce MTL into the LSTMs approach.

This avenue was pursued, but was unsuccessful. This was mostly due to the fact that in most of the datasets used in the MTL work (Table 3.1), the character-level attention LSTM was outperformed by the single-task CNN model. We hypothesized that this was because most of the other datasets are quite small and thus do not provide enough data for the LSTM to learn well. In smaller datasets, the convolution filters (which is an approximation of an *n*-gram window) can become more capable of identifying the features of named entities from the surrounding context words than a sub-optimally trained LSTM with character-level features.

3.7 Conclusion

NER is an important part of LBD and knowledge discovery. Various recent advances in neural networks have opened new avenues for improved NER. These include semantically-rich word embeddings, MTL with CNNs, and character-level features with LSTMs. Seeking to exploit these methods, we did work on three aspects of NER with the aim of improving its performance for biomedical NER. We worked on extrinsic evaluation of improved biomedical word embeddings as inputs to neural NER models, MTL with CNNs to harness the information in disjoint datasets and on attention-based, character-level sequence labelling for NER.

The word embeddings developed were evaluated extrinsically, using a CNN NER model we developed, on two biomedical NER datasets. They led to improved performance over the existing embeddings. During this testing, we found divergent results from changing the value of the *window size* parameter in the embedding creation model. All sets of vectors showed a notable increase in the intrinsic evaluation scores when this parameter was increased. However, the extrinsic evaluation showed that all results in extrinsic tasks have

peak performance when the parameter was at its lowest value, followed by a gradual decrease as the parameter increased.

For investigations into MTL, we first developed a single-task CNN model for NER and then two variants of a multi-task CNN. We trained these on 15 domain-specific datasets representing a wide range of biomedical named entities. We observed an average improvement on MTL in comparison with single-task learning. Individually, there were also significant improvements on many of the datasets. Although there was a drop in performance on some tasks, for most tasks performance improves significantly. We also found that MTL is beneficial for small datasets. Across the various settings the improvements are significant, demonstrating the benefit of MTL for biomedical NER.

The investigations into character-level extensions to models for sequence labelling tasks such as NER led to a novel architecture for combining alternative word representations. By using an attention mechanism, the model is able to dynamically decide how much information to use from a word- or character-level component. We evaluated different architectures on a range of sequence labelling datasets, and character-level extensions were found to improve performance on every benchmark, especially the biomedical NER datasets. In addition, the proposed attention-based architecture delivered the best results.

Once concepts are extracted from biomedical texts as named entities, they can be used for LBD. In our case this means making them nodes in a graph with various relationships and using the information there to perform LBD. If the graph is created such that the information it represents is the state of the biomedical literature at the time it was created, any additional links in the graph formed at a later time period will represent an addition to the literature and thus is a candidate for LBD. The field of *link prediction* is concerned with inferring missing links from a graph so it has potential to be used for LBD. Exploring the ways in which this can be done is the topic of the next Chapter.

Chapter 4

Leveraging Link Prediction

4.1 Introduction

Literature-Based Discovery (LBD) is concerned with both filling in gaps in knowledge, as represented in literature, and extending it. This knowledge can be represented by graphs and lots of biomedical knowledge are explicitly represented as graphs. Filling gaps in and extending graphs is the field of *link prediction*, thus link prediction can be useful for LBD. in fact, LBD can be fashioned as constrained link prediction: it is link prediction restricted to links between two nodes A and C where there exists a set of B nodes on a (usually restricted) path connecting A and C.

In the context of LBD, link prediction can involve predicting connections between biomedical concepts where such connections do not currently exist in the literature. It can be used for LBD by predicting the contribution of future discoveries in the field and for predicting missing information in biomedical knowledgebases. Using link prediction for filling in missing information without taking into account the time dimension can be used for completion, and when the evolution of the graph over time is taken into account it can highlight potential future discoveries, which is directly analogous to LBD. We did work which investigated using neural networks for link prediction in realistic biomedical graphs for several tasks including LBD (Crichton et al., 2018).

4.2 Link Prediction in Biomedical Data

Link prediction is the task of predicting edges between nodes in a graph which are currently not present in the graph. Liben-Nowell and Kleinberg (2003) first formulated the link prediction problem in social networks. Most link prediction works have focused in large

part on determining which links will form next in various types of social networks where links can represent friendships (Backstrom and Leskovec, 2011; Leskovec et al., 2010), collaborations and co-authorships (Al Hasan et al., 2006; Backstrom and Leskovec, 2011), citations (Benchettara et al., 2010) and online transactions (Benchettara et al., 2010) among others. Additionally, link prediction has been used on large-scale knowledgebases to add missing data and discover new facts (Nickel et al., 2016; Schlichtkrull et al., 2018). One of those large-scale knowledgebases, Knowledge Vault (Dong et al., 2014), used an MLP to perform link prediction as we do in this work.

Link prediction has already been applied in the biomedical domain for various uses including LBD. Katukuri et al. (2012) used Decision Trees and a Support Vector Machine (SVM) to perform supervised link prediction on a large-scale biomedical network of concept co-occurrence in documents to generate hypotheses. They used topological as well as semantic features to predict links which represented cross-silo hypotheses in a literature-sliced corpus. Predicting Drug Target Interactions (DTIs) is important in repositioning existing or abandoned drugs by identifying new uses for them. Wang and Zeng (2013) and Lu et al. (2017) both used link prediction on this task by providing *in silico* predictions of interactions. Wang and Zeng (2013) used Restricted Boltzmann Machines (RBMs) to predict different types of DTIs on a multi-dimensional network while Lu et al. (2017) used similarity indices to predict links in DTI networks.

Connections which indicate relationships between concepts is the second vital piece of information which is needed to perform LBD. The presence of these connections formalizes some level of relatedness with certainty between the concepts involved which distinguishes them from a pair of concepts which have no connection. Popular approaches for obtaining these connections include: co-occurrence (Preiss et al., 2012); relationships from external knowledgebases, such as protein-protein interactions; gene-disease associations (Eronen et al., 2012) and existence of established real-world relationship such as between papers and their authors (Sebastian et al., 2015).

4.3 Biomedical Graphs

The biomedical domain has a wealth of datasets which encapsulate varied and useful information which can be represented as graphs, and many already are. It is useful to know if any information is missing from these or what information may be added to them in the future. Since link prediction is the task of proposing links which are not currently part of a graph but could become a part of it, if the information in these datasets are represented as graphs, link

prediction has application in various biomedical information processing tasks. These include predicting Drug-Target Interactions (DTI) for drug re-purposing, predicting Protein-Protein Interactions (PPI), facilitating Literature Based Discovery (LBD) for generating hypotheses from publications and automating knowledgebase completion. Much of this depends on what the links in the graph represent. This determines what data the graph contains and what is the significance of the links predicted.

Link prediction has been used for predicting DTI by applying it to graphs representing drugs/chemicals and the proteins which they interact with (Lu et al., 2017; Wang and Zeng, 2013). It has also been used to facilitate LBD by applying it to bibliographic networks (Katukuri et al., 2012; Sebastian et al., 2015) and term co-occurrence networks (Preiss et al., 2015). Kastrin et al. (2016) also used it on MeSH to demonstrate its use on graphs of organised knowledge. Grover and Leskovec (2016) used it to predict PPI from a subset of the BioGRID graph (Stark et al., 2006).

4.4 Embedding Graphs

Many of the existing approaches to link prediction do not make use of the information contained in the structure of the graph, which can aid in link prediction. Others which do use this information either do so using approaches which are only able to draw a limited amount of patterns from the graph or provide restricted datasets to their methods. Our work on this task made use of information in the graph by using methods which are able to extract non-linear patterns from graph structures and use this information to predict the likelihood of a link forming between two nodes.

This is possible in large part to the recent rise in the number of works using various neural network approaches to embed graphs in low-dimensional spaces. These produced vectors of real numbers which are representations of a graph's nodes that aim to place similar nodes close to each other in the vector space and dissimilar ones far apart based on the structure/topology of the graph and in some cases the weights of the edges. These node vectors are thus similar to the word embeddings explained in Section 3.3 and are called *node embeddings* to reflect this. The earliest methods that create them include DeepWalk (Perozzi et al., 2014), node2vec (Grover and Leskovec, 2016), LINE (Tang et al., 2015), SDNE (Wang et al., 2016) and HOPE (Ou et al., 2016), although many more with various benefits appeared later.

These node embedding creation methods opened the possibility of training rich representation as inputs to neural link predictors which output how likely it is for a link between two nodes to form. Several works have already begun to explore this avenue and report promising

results, however their approaches have not comprehensively addressed the issues of using these methods for link prediction. Particularly lacking are experiments in realistic settings like time-slicing, where graphs are split so that predictors are evaluated on how well they predict chronologically later links; and evaluating performances with metrics where all nodes have equal weight, since link prediction applications usually need to perform well across a wide cross-section of nodes as opposed to performing very well on few nodes which are hubs in the graph and are thus easy to perform well at. These investigations are pertinent if link prediction is to be used for LBD. Time-slicing is necessary as knowledge discovery proceeds chronologically and LBD systems should be capable of performing well across a large number of nodes instead of very well at a few hub nodes and poorly otherwise. This is because LBD is inherently a node-centric task: knowledge is being discovered about particular entities of interest to the end user, not across the entire field of knowledge.

Graphs encode knowledge and can be processed to extract information which may not be easily seen before. For a machine to perform this processing, the graph must be represented in a format which it can use, usually by representing nodes as vectors of real numbers. Works on node representation aim to devise methods which can create vector representations which preserve the original information in the graph. In general the information in a graph can be classified as first or second (and higher) order proximity (Goyal and Ferrara, 2018b; Tang et al., 2015).

Given two nodes in a graph, first order proximity is concerned with the strength of the direct link between them. Second order proximity between two nodes compares their neighbourhoods and classes them as similar if their neighbourhoods are similar. The extent to which a method can preserve the proximities of a graph when creating representations determines its quality. The node representations created by recent research models each node as a vector in a space where similar nodes are located close to each other. Figure 4.1 uses t-SNE (Maaten and Hinton, 2008) to visualise a portion of 2-dimensions of an example of such a vector space for the PubTator dataset used in this work created with the DeepWalk method.

Since there has been a proliferation of methods which seek to create these node embeddings from graphs and it would be unwieldy to include all of them in this work, we investigate four of the most popular ones whose implementations are freely available online.

- **DeepWalk** (Perozzi et al., 2014) uses random walks on graphs to learn latent representations of nodes and encodes them in a continuous space. It does this by treating random walks on graphs like sentences in a natural language and generalizes recent advancements in language modelling (Mikolov et al., 2013a) developed for word sequences to graphs. This makes it easy to use existing language modelling tools to

implement, but it consequently lacks an objective function which explicitly captures the graph's structure. As a result of using these language modelling tools, its vector space take on the characteristics of word vector spaces where words which appear in similar context are embedded close to each other, likewise nodes which appear in similar neighbourhoods would be deemed similar and embedded close to each other.

- **Large-scale Information Network Embedding (LINE)** (Tang et al., 2015) explicitly defines two optimization functions to capture the structure of the graph. One captures first order proximity and the other captures second order proximity. As with the other methods, this results in nodes which are explicitly connected and which share similar neighbourhoods being deemed similar. They report that training their model with each setting then concatenating the outputs gave the best performance.

- **Node2vec** (Grover and Leskovec, 2016) is similar to DeepWalk in how it preserves higher order proximity between nodes. It does so by maximizing the probability of the occurrence of subsequent nodes in random walks over a graph, resulting in nodes which share similar environments being deemed similar. The difference to DeepWalk is that node2vec's random walks are parametrized to provide a trade-off between prioritising breadth-first or depth-first walks. This allows for similarity to be defined not only based on the neighbourhood similarity of two nodes but also on the role they play in the network. For example, walks which prioritise breadth-first search will capture the similarity of two nodes which are in the same neighbourhood while those which prioritise depth-first search can capture the similarity of two nodes which are in different immediate neighbourhoods but play the role of hub node in both neighbourhoods. Choosing the right balance enables node2vec to preserve first- and second-order proximity between nodes to potentially produce more informative walks, leading to superior embeddings.

- **Structural Deep Network Embedding (SDNE):** Wang et al. (2016) argue that the shallow models which the other methods use cannot adequately capture the highly non-linear structure of most graphs. Since deeper models have proven successful at capturing non-linearity in complex data, they use them to create representations. Their model jointly optimises unsupervised and supervised parts. The unsupervised part produces an embedding for a node which can reconstruct its neighbourhood. The supervised part applies a penalty when nodes deemed to be similar are mapped far

Fig. 4.1 Visualisation of a subset of the vector space created by DeepWalk from the PubTator dataset. Vectors of nodes representing respiratory infections are close to 'Viral Pneumonia' while those of acids and chemicals are close to 'Hydrochloric Acid' indicating that their vectors are more similar.

from each other in the vector space. Here as well similarity means a combination of nodes with explicit connections as well as nodes which share similar neighbourhoods.

4.5 Node Embeddings for Link Prediction

There have been recent works which used the embeddings created from neural network methods for link prediction. The evaluation metrics mentioned here are explained in Section 4.6.1. To the best of our knowledge, none of these works included time-sliced datasets and the sizes were generally smaller than realistic biomedical graphs.

Grover and Leskovec (2016) evaluated node2vec embeddings on three graphs, including a PPI subset of BioGRID, and compared the results to existing well-known and competitive

link prediction metrics Common Neighbours, Jaccard Index, Adamic-Adar and Preferential Attachment. This work evaluated using Area Under the Receiver Operator Characteristics Curve and its largest graph contained 19,706 nodes and 390,633 links.

Wang et al. (2016) used the embeddings created from SDNE on a single dataset of 5,242 nodes and 28,980 links. They compared to LINE, DeepWalk, GraRep, Laplacian Eigenmaps and Common Neighbours. They evaluated using precision at k for the full network and Mean Average Precision (MAP) for a sparse version of the graph.

Ou et al. (2016) performed link prediction on two graphs to compare performance of HOPE to Partial Proximity Embedding, LINE, DeepWalk, Common Neighbours and Adamic-Adar. The larger graph had 834,797 nodes and 50,655,143 links. They randomly sampled 0.1% of node pairs for evaluation but the amount used for creating embeddings is not reported. They evaluated using precision at k.

Goyal and Ferrara (2018b) compared the performances of Laplacian Eigenmaps, Graph Factorization, node2vec, SDNE and HOPE to perform link prediction on four datasets including a PPI subset of BioGRID. They evaluated using precision at k and MAP to determine how performance corresponded to changes in vector dimensions. They experimented on five random subsets of each graph created such that each subset contained 1,024 nodes.

4.6 Link Prediction with Neural Networks and Node Embeddings

In this part of the work we employed four graph embedding algorithms: DeepWalk, LINE, node2vec and SDNE. We investigated how a neural predictor, using representations from these methods, performs on link prediction in biomedical graphs containing information which can be used for several bioinformatics tasks including DTI, PPI and LBD. We compared this approach to the performance of established baseline methods Common Neighbours, as used in (Newman, 2001); Adamic-Adar (Adamic and Adar, 2003) and Jaccard Index (Jaccard, 1901). These methods were chosen because they continue to be very competitive and challenging baselines for link prediction (Liben-Nowell and Kleinberg, 2003; Wang et al., 2016), are conceptually simple and scale well to large graphs which are of realistic sizes in the biomedical domain and useful for LBD.

We report results on graphs which represent real biomedical information in settings where links were randomly removed as well as where links were removed by time-slicing. These results show evaluations with metrics that weigh the performance at each node equally and those which do not as they illustrate different aspects of a predictor's performance and can

be useful depending on its application. These contributions together provide large-scale comparisons and analyses that inform and explain the best approaches to link prediction using neural networks with node embeddings and highlight areas of further research.

4.6.1 Important Considerations in Link Prediction

This section presents some factors which affect link prediction experiments and thus the interpretability and applicability of their results. To the best of our knowledge, no study preceding this one using node embeddings and neural networks for link prediction took all of these factors into consideration.

Link Prediction Setting

There are two main link prediction settings: random- and time-slicing. In random-slicing, a percentage of the links are removed randomly and evaluation consists of predicting the removed links. This can be useful for filling in missing information such as gaps in the literature without a time component. Time-slicing (or literature-slicing) aims to take the temporal evolution of the graph into account and only links formed after some point in time, t, are removed. The state of the graph before t is given to the link predictor and its aim is to predict links formed at a later time.

The first setting is applicable when the current knowledge represented by the graph is incomplete and link prediction aims to complete it as well as when the temporal data for the graph is unknown or irrelevant. The second can be used to predict the future state of the graph and so can suggest feasible links to investigate. This setting can make link prediction more challenging for two reasons:

1. New nodes can be introduced to the graph at later time periods which will present little or no information to the link predictor to use as these nodes will have no links to other nodes in the time period which the predictor uses to make predictions,

2. In evolving graphs, the easier links tend to form before more difficult ones, so the links to be predicted in later time periods tend to be more difficult. This evolution is akin to hypothesis testing and generation which LBD explicitly aims to perform.

Meaningful Evaluation Metrics

Several metrics which measure different aspects of a predictor's performance have been used to evaluate link prediction methods. It is useful to distinguish between metrics which weigh performance at all nodes in the network equally and metrics which do not. We refer to the

former as an node-equality metrics and the latter as link-equality metrics. Node-equality metrics can be robust to performance at hub nodes, which tend to be easier for link prediction, and some link prediction applications are more concerned with how a predictor performs across a cross-section of nodes than how many links it predicts across the entire graph. This is analogous to the difference between micro- and macro-averaging. Node-equality metrics can be particularly useful for LBD as it is concerned with a method's performance across a wide cross-section of nodes because hypothesis testing and generation are inherently node-centric. Prior work involving neural link prediction mainly reported evaluations on link equality metrics.

The following five metrics were used in this and previous works. In-depth explanations of these metrics can be found in several works including (Goyal and Ferrara, 2018b; Yang et al., 2015). For all the metrics, a higher score indicates better performance. It is useful to define some terms to understand the definition of the metrics.

- True Positives (TP): The links which a predictor predicts as positive and actually are true missing or future links.
- False Positives (FP): The links which a predictor predicts as positive but actually are not missing or future links.
- True Negatives (TN): The links which a predictor predicts as negative (or not existing) and actually turn out to not be missing or future links.
- False Negatives (FN): The links which a predictor predicts as negative (or not existing) but which turn out to be missing or future links.
- Recall or true positive rate = $\frac{|TP|}{|TP|+|FN|}$
- Precision = $\frac{|TP|}{|TP|+|FP|}$
- Fallout or false positive rate = $\frac{|FP|}{|FP|+|TN|}$

1. **Area under the Precision-Recall Curve:** Recall measures what percentage of all positives were returned. Precision measures what percentage of the results are true positives. These metrics are used to construct a Precision-Recall Curve which illustrates how the increase in recall affects precision. The area under this curve can be used to evaluate link prediction. When precision and recall are not restricted to involve a particular node, this is a link-equality metric.

2. **Area Under the Receiver Operating Characteristics Curve:** True positive rate is equivalent to recall. The fallout or false positive rate measures how many negatives were returned as false positives by the predictor. These metrics are used to construct a Receiver Operating Characteristics (ROC) Curve which illustrates this relationship. The area under this curve is used to evaluate link predictors. When the true and false positive rates are not restricted to involve a particular node, this is a link-equality metric.

3. **Precision at k:** The above metrics measure performance across all recall levels but some uses of link prediction are only interested in the quality of highly ranked results. Precision at k or the top k predictive rate is the percentage of true positives among only the top k ranked links. This is usually used to return the top k of all possible links thus it is a link-equality metric.

4. **Mean Average Precision (MAP):** Given a ranked list of predicted links relevant to a particular node, we can calculate the precision after each true positive. The average of these values gives the average precision for that node. This done over all nodes in the graph gives a single value, node-equality measure.
   $MAP = \frac{\Sigma_i AP(i)}{|V|}$,
   where $|V|$ = number of nodes, AP(i) = $\Sigma_n(R_n - R_{n-1})P_n$ and P_n and R_n are the Precision and Recall at the n^{th} threshold for the i^{th} node.

5. **Averaged R(elevant)-Precision:** Similar to MAP but instead of calculating the precision after each positive link in the list of results for a given node, precision is only calculated with the top R results. R is determined by how many true positives exist for the node. The main difference from MAP is that this metric does not consider the remainder of the ranked list outside of the length of the top R. This also gives a single value, node-equality measure. This metric is similar to precision at k except that instead of having a fixed k, it changes based on the amount of positives each node has so that a node with less than k positives is not penalised and a node with a lot more positives than k is not easier for the predictor to perform well at.

   Averaged R-precision = $\frac{\Sigma_i Pr@R(i)}{|V|}$,
   where $|V|$ = number of nodes, $Pr@R(i)$ = precision at R for the i^{th} node with R positives.

Scalability, sparsity and negatives

Biomedical and other real-world graphs reflect complex relationships between numerous entities so to be truly useful, methods employed to make use of them should be able to scale, usually to hundreds of thousands of nodes and millions, or billions of links.

Supervised machine learning approaches require both positive and negative examples to train models. Negatives are created from links which do not exist in the network. Graphs tend to be sparse as only a small fraction of potential links are actually formed. While a link between two nodes in a graph confirms a relationship, the absence of a link does not confirm a lack of relationship thus the assumption that most node pairs which do not have a link have no relevant relationship is not always true. As a result of this, links can potentially be used as negative examples in supervised machine learning techniques for link prediction which should not be negatives, because they will in fact form later. In real-world situations, the model will inevitably encounter such links and it will be trained on some negative examples which would later turn out to be positive.

Due to the problems of large size and extreme sparsity, it is usual to create negatives for training and testing by sub-sampling from the list of potential negative links. The manner in which this sub-sampling is done can affect the performance of the link predictor. Yang et al. (2015) looked in great detail into these issues and how they can affect link prediction evaluation. The issue of scalability also affects the ratio of negative to positive examples in the evaluation data. In real-world situations the number of unformed links far outweigh the formed ones, but it is often computationally prohibitive to replicate the real positive to negative ratio or to even approximate it for large graphs.

Node combination method

A neural network approach to link prediction with node embeddings requires the model to represent the input link as a single vector so the embeddings of the nodes involved in a link need to be combined. This can be done in several ways which can affect the predictor's performance. Concatenating the embeddings is simple and preserves all information but increases the size of the input vector in proportion to the amount of nodes and relationships comprising the link. Grover and Leskovec (2016) used four methods which give a constant input size and we experimented with these in addition to concatenation. These are detailed in Table 4.1.

Operator	Definition
Average	$\frac{f_i(u)+f_i(v)}{2}$
Concatenate	$f(u)$. $f(v)$
Hadamard	$f_i(u) * f_i(v)$
Weighted-L1	$\|f_i(u) - f_i(v)\|$
Weighted-L2	$\|f_i(u) - f_i(v)\|^2$

Table 4.1 Node Combination methods on vectors of nodes u and v. Binary operators operate on the i^{th} element.

4.7 Experimental Methods

This Section gives in-depth details about how the experiments were conducted.

4.7.1 Datasets

The graphs we use were created from the following datasets. The graph details can be found in Table 4.2.

Biological General Repository for Interaction Datasets (BioGRID): This is an open database created from manually curating experimentally-validated genetic and protein interactions that are reported in peer-reviewed publications (Stark et al., 2006). The latest major release (Chatr-aryamontri et al., 2017) includes over 1 million Genetic and Protein interactions across all major organism species and humans. Links in this graph represent interactions between biomedical entities derived from published, experimentally-validated genetic and protein interactions, including PPI. We used version 3.4.147 of this dataset.

Manually Annotated Target and Drug Online Resource (MATADOR): This is an open online DTI database (Günther et al., 2008). It includes interaction between chemicals and proteins. Following Lu et al. (2017) the Chemical and Protein IDs are used to form a bipartite DTI graph. Thus the links in this graph represent interactions between chemicals and proteins representing drugs and targets respectively.

PubTator: Biomedical entities recognised by PubTator (Wei et al., 2013) mentioned in the titles and abstracts of PubMed publications from 1873 to 2017 were used to create this dataset. A link exists between two biomedical entities if they co-occur in a single sentence. The annotations were downloaded on June 20th, 2017.

4.7.2 Settings for Training Node Representation Methods

The hyper-parameter settings for DeepWalk and LINE were the same as used in (Wang et al., 2016) which is a recent work which compared both of those methods. Parameters

Dataset	Node Count	Link Count	Has Dates	Link Type
BioGRID	65,026	1,076,308	Yes	Published Interactions
MATADOR	3,704	15,843	No	Drug-Target Interactions
PubTator	265,148	6,854,054	Yes	Literature Co-occurrences

Table 4.2 The datasets and their relevant details. The link counts here are of undirected links.

for node2vec which overlapped with DeepWalk's were set to the same values. All methods created embeddings of 100 dimensions as this was determined to be a good value on datasets which are not used as part of this work.

DeepWalk: window size = 10, walk length = 40, walks per vertex = 10. **LINE:** learning rate = 0.025, number of negative samples = 5 and total number of samples = 10 billion. According to (Tang et al., 2015), LINE performs best when it is run twice to obtain first- and second-order proximity embeddings which are concatenated and L2 normalized. We follow their recommendations. For each order we created half the number of dimensions as needed so that when they were concatenated, the final result had the appropriate number. **node2vec:** window size, walk length and walks per vertex were the same as DeepWalk's. The parameters p and q were 2 and 4 respectively as randomly chosen from the optimal set given by the creators (Grover and Leskovec, 2016). We used **SDNE** implementations from both (Goyal and Ferrara, 2018b) and (Wang et al., 2016) with hyperparameters as used by (Goyal and Ferrara, 2018b): α = 1e-6, β = 5, ρ = 0.3, *xeta* = 1e-4 and *nu1 & nu2* = 1e-3.

4.7.3 Neural Link Predictor and Baselines

The neural link predictor was a binary classifier implemented as a Multi-layer Perceptron (MLP) neural network with a single hidden layer containing 100 Rectified Linear Units (ReLU) (Nair and Hinton, 2010). It accepted the vector representation of two nodes representing a link by combining their individual vector representations with operators defined in Table 4.1 and output the probability of a link forming between the nodes. These probability scores were used to create a ranked list of all links in the evaluation set. The model was trained for 7 epochs. This minimalist model was chosen so that the contribution from each node embedding method could be compared without the confound of the contribution of a powerful neural network model, although a powerful neural network model can also be used. The other parameters were determined to be a good values based on datasets which are not used as part of this work.

We employed three baseline methods which have been used successfully for link prediction: Adamic-Adar, Common Neighbours and Jaccard Index. It is necessary to modify these slightly for bipartite graphs following (Huang et al., 2005). Their definitions are in Table 4.3.

Name	Definition	Bipartite Definition
Adamic-Adar	$\frac{1}{log(\lvert N(u) \cap N(v) \rvert)}$	$\frac{1}{log(\lvert N(u) \cap \hat{N}(v) \rvert)}$
Common Neighbours	$\lvert N(u) \cap N(v) \rvert$	$\lvert N(u) \cap \hat{N}(v) \rvert$
Jaccard Index	$\frac{\lvert N(u) \cap N(v) \rvert}{\lvert N(u) \cup N(v) \rvert}$	$\frac{\lvert N(u) \cap \hat{N}(v) \rvert}{\lvert N(u) \cup \hat{N}(v) \rvert}$

Table 4.3 Baseline methods for node pair *(u, v)* with neighbour sets *N(u)* and *N(v)*. $\hat{N}(x)$ are the neighbours of the neighbours of x.

4.7.4 Experiments

We experimented with both link prediction settings explained in Section 4.6.1 where possible. For the MATADOR dataset, there was no temporal data so no time-sliced experiments could be done. The existing links of each graph were split into 3 segments.

For the random-slice experiments, 60% of the links were used to create the node embeddings, which included 10% used to train the neural link predictor where necessary and the remaining 40% were used to evaluate the predictors. The data used to train the model was also used to induce the embeddings since there is no reason to withhold that information from the node representation methods and more information will lead to better representations. The test set is larger than is usually found in machine learning works but being able to demonstrate good results with reduced training data is generally a desirable quality. For time-slice experiments, we sought to have similar split sizes as the random-sliced, but exact sizes were not possible since this is dependent on the amount of links in a year. The details of the time slices are in Table 4.4.

For both settings, after splitting the existing links, we then sub-sampled negative examples by randomly sampling from all the possible node pairs without a link while maintaining a 1:1 ratio of positive to negative links. Following (Grover and Leskovec, 2016), graph connectivity was maintained in the random-sliced data, but this was not possible to enforce in the time-sliced data as the links in each slice were determined by what year they were added to the dataset. Due to the varying sizes of the graphs, for precision at k we let the total amount of positives which can be returned dictate the k. We report k to be 30% of all possible positives here. Results on additional k values can be found in Appendix B. We implemented the baselines listed in Section 4.7.3 and used them on the same induction, train and evaluation subsets. We used Scikit-learn (Pedregosa et al., 2011) to efficiently calculate most of the metrics on the predictions of the neural and baseline link predictors.

Dataset	Link Use	Time Slice	Link Count	Link Percentage (%)
BioGRID	Induction	1970-2014	678,994	63.08
	Train	2013-2014	121,442	11.28
	Test	2015-2017	397,302	36.91
PubTator	Induction	1873-2003	4,069,683	59.38
	Train	2001-2003	614,031	5.90
	Test	2004-2017	2,784,371	40.62

Table 4.4 Time Sliced details. Induction includes Train

4.8 Results and Discussion

The scores presented in the result tables are the means of three runs of each experimental setting. Scores in **bold** represent the best score for a particular metric. The best score and all other scores were tested for statistical significance using a two-tailed t-test with $\alpha = 0.05$. Scores with an asterisk (*) are not significantly different from the best score, scores without an asterisk are significantly different. The standard deviation of the means reported here were excluded to aid readability but can be found in the full result tables in Appendix B.

The performance of the neural classifier with inputs combined using Hadamard, Weighted-L1 and Weighted-L2 are not the best performers in any experiments so they are left out of the tables in this Section. The results for embeddings created with SDNE are much poorer than the others and are left out of these tables for brevity. The full set of results containing all these figures can be found in Appendix B. It also contains analysis about interesting results involving DeepWalk embeddings combined with Weighted-L1 and -L2. The most efficient reference implementations of SDNE available exceeded the available computational resources for the BioGRID and PubTator graphs, so we report no results for them in those settings.

4.8.1 MATADOR

These results are in Table 4.5. The Common Neighbours and Jaccard Index baselines are the best performers across all metrics. This can be attributed to the graph being too small for the neural network methods to create good embeddings for each node which lead to poor input to the neural link predictor. For precision at k, averaged and concatenated DeepWalk embeddings also produce comparable results. Adamic-Adar performs the worse of the baselines despite the fact that it is common neighbours-based. This is because the algorithm weighs a small amount of shared items between entities high and a higher amount of shared items less. As we are only using amount of common neighbours as the shared item

Method	Node Combination	AUC (ROC)	AUC (PR)	MAP	Avg. R-prec	Prec @ k
Deep-Walk	Average	95.93	95.82	89.81	86.86	98.77*
	Concat	94.97	94.83	88.30	84.63	98.34*
LINE	Average	80.63	81.30	67.74	61.04	91.65
	Concat	81.16	81.82	68.53	61.42	92.00
node-2vec	Average	78.38	78.75	66.42	59.32	88.67
	Concat	77.62	77.54	65.44	58.40	87.25
AA	N/A	91.97	88.40	87.16	85.06	86.87
CN	N/A	**97.27**	97.04*	**95.47**	**94.64**	98.74*
JI	N/A	97.23*	**97.10**	94.72	92.29	**98.96**

Table 4.5 MATADOR random-slice results

between two nodes here, links which score high for common neighbours will score lower for Adamic-Adar.

4.8.2 BioGRID

Random-slice: The results of this experiment are in rows 4-12 (the top half) of Table 4.6. Concatenated and averaged node2vec embeddings are the best performers across 4 of the 5 metrics and the best performer in the remaining metric is not significantly better. Averaged LINE embeddings are not significantly different from the best performer in any metric. In general the neural network approaches outperform the baselines. This is not surprising as there are favourable conditions for the neural network methods: there is a large amount of data to induce the node embeddings with and, since connectivity is guaranteed, all nodes have a chance of getting an embedding which is better than its random initialization. These embeddings would then perform better in the neural link predictor.

Common Neighbours is the best performer for precision at k, although it is not significantly better than four neural network approaches. The chosen k focuses only on the very highly ranked links and other works have already posited that Common Neighbours returns good results at the top of its ranked list (Lu et al., 2017). Its failure to perform well for the AUC metrics highlights that lower in its ranked list of links, performance degrades substantially. Its poor performance at the node-level metrics also indicate that the links which it is predicting correctly at the top of its ranking are dominated by the links of hub nodes.

Time-slice: These results are in rows 15-23 (the bottom half) of Table 4.6. Averaged node2vec embeddings are the best performer for three of the metrics and embeddings combined by concatenation are not significantly worse in two of the metrics. Common Neighbours performs the best in two metrics, including one node-level metric where it is

significantly better than all other approaches. In general, the performances of Common Neighbours and Jaccard Index are not as far behind that of the neural network approaches as they are for the random-sliced setting of this dataset. This is due to a property of the dataset: it is skewed towards later publications. Because of this bias, when it is time-sliced as detailed in Table 4.4, 14.5% of the nodes representing entities in the test slice had never occurred in the induction slice. The neural network approaches could not create good embeddings for these nodes so they are simply assigned their randomly initialized values, which contain no useful information and so negatively influenced the neural link predictor's performance. This is an instance of new nodes appearing in the evaluation slice of the graph as mentioned in Section 4.6.1 and makes a difference to performance, which highlights the importance of evaluating by time-slicing.

It is interesting that the best performer for each of the node-level metrics is different and the difference between them is significant in each case. This indicates that the neural predictor using averaged node2vec embeddings is good at ranking true positives for a given node within the top R while Common Neighbours is better at ranking more positives at the very top of the lists but is unable to do so for some positives. Based on the performance on the same graph random-sliced, it may be that the more difficult nodes are the ones it fails to perform well at.

4.8.3 PubTator

Random-slice: These results are in rows 4-12 (the top half) of Table 4.7. Concatenated DeepWalk embeddings produce the best results in three of the metrics and is not significantly worse in another. Averaged and concatenated LINE embeddings are on par with the best results except in a single instance.

An interesting result is the dual observation that Common Neighbours performs the best for averaged R-precision while its performance for MAP is significantly worse than the best. Taken together, these indicate that it captures several true positives for a given node within the top R but not rank them at the top of that list and is prone to ranking some of the true positives quite low. The approaches which outperform it for MAP but not for averaged R-precision are better at ranking true positives just outside of the top R than it is.

Note that this is the reverse of what we found for Common Neighbours on the time-sliced BioGRID dataset. Both the setting and information encoded in the graphs are different and this illustrates that these can have an impact on link predictor performance and need to be taken into consideration when drawing conclusions from link prediction experiments.

Time-slice: These results are in rows 15-23 (the bottom half) of Table 4.7. Similar to the random-sliced experiments on this dataset, concatenated DeepWalk vectors produce the

Method	Node Combination	Random Slice AUC (ROC)	AUC (PR)	MAP	Avg. R-prec	Prec @ *k*
Deep-	Average	97.69	97.62	79.24	73.86	99.30
Walk	Concat	97.74	97.65	82.48	77.70	99.18
LINE	Average	98.10*	97.80*	83.13*	78.22*	99.54*
	Concat	98.08	97.76	82.94	78.04	99.29
node-	Average	98.32*	97.97*	85.70*	81.17*	99.38*
2vec	Concat	**98.51**	**98.26**	**86.49**	**81.84**	99.49*
AA	N/A	86.10	90.75	70.97	57.65	96.13
CN	N/A	91.20	94.96	75.72	69.81	**99.64**
JI	N/A	90.80	93.95	73.93	68.79	98.59
		Time Slice				
		AUC (ROC)	**AUC (PR)**	**MAP**	**Avg. R-prec**	**Prec @ *k***
Deep-	Average	89.40	90.10	68.94	63.30	97.25*
Walk	Concat	92.12	92.78	71.61	65.96	98.04
LINE	Average	91.86	92.31	72.85	67.76	97.40
	Concat	93.55	93.74	73.60	68.57	97.90
node-	Average	**95.25**	**95.43**	74.91	**70.39**	98.26
2vec	Concat	93.66	94.66*	73.48	68.77	98.40*
AA	N/A	77.46	87.69	74.84	61.39	98.10
CN	N/A	85.07	91.81	**76.20**	67.73	**99.38**
JI	N/A	84.74	90.20	75.60	67.49	97.45

Table 4.6 BioGRID random-slice and time-slice results

Method	Node Combination	Random Slice AUC (ROC)	AUC (PR)	MAP	Avg. R-prec	Prec @ *k*
Deep-Walk	Average	98.85	99.01	83.67	75.97	99.93*
	Concat	**99.20**	**99.30**	**91.01**	85.46	99.94*
LINE	Average	99.10*	99.23*	90.36*	84.56	**99.97**
	Concat	99.13	99.24	90.07	84.03	99.95*
node-2vec	Average	98.71	98.90	82.98	75.29	99.94*
	Concat	99.16	99.21	88.94	82.14	99.92*
AA	N/A	92.92	84.56	56.48	66.38	83.33
CN	N/A	98.40	98.28	79.84	**87.10**	99.94*
JI	N/A	92.36	87.59	65.44	59.74	91.21
		Time Slice				
		AUC (ROC)	**AUC (PR)**	**MAP**	**Avg. R-prec**	**Prec @ *k***
Deep-Walk	Average	93.86*	95.51*	70.78*	62.16*	**99.89**
	Concat	**93.99**	**95.70**	**71.11**	**62.65**	**99.89**
LINE	Average	88.68*	92.27*	55.61*	46.41*	**99.89**
	Concat	90.32	93.01	62.51	53.21	**99.89**
node-2vec	Average	88.40	92.07	55.72	46.48	99.87
	Concat	88.13	91.83	53.24	43.69	99.84
AA	N/A	85.10	80.24	35.49	40.13	90.56
CN	N/A	88.37	88.83	43.67	46.59	99.84
JI	N/A	86.08	83.52	38.66	38.75	94.27

Table 4.7 PubTator random-slice and time-slice results

best results in all metrics although there is a four-way tie for precision at k. Averaged LINE embeddings are on par with the best results here as well. The neural network approaches vastly outperform the baselines. Although this graph only contains co-occurrence information, this is noteworthy as this is the largest graph, in a difficult realistic setting and with no apparent biases to hinder the neural network methods. It is of particular importance to this thesis as node-centric performance is of more importance for LBD and co-occurrence information has been used extensively for LBD in the literature.

4.8.4 General Discussion

Investigating nodes with no common neighbours

We hypothesize that the superior performance of the neural network methods are due to the limitations in recall of Common Neighbours and baselines based on it. It is possible for links to form between nodes which have no previous common neighbours and these

methods would fail in such cases. We investigated this limitation and the effect it has on the performance of the link predictors. We first quantified these links in the test examples of each experimental setting then looked at how the best predictors in each category ranked these links. In the latter, we specifically looked at whether the links were ranked in the top or bottom half of the overall ranked lists. Since there are equal number of positive and negative links, a good predictor would rank a high amount of links in the top half. The neural network approaches performed vastly better in those cases, although the varying amount of such positives affected the overall effect.

In the following discussion, it is important to bear in mind that for links which have no prior neighbours, the baselines would all assign them a score of zero, so any which appear in the top half will do so by pure chance as they would be tied with all the others (positive and negative) which also score zero. These can be thought of as links in the evaluation data which the baselines had no chance of getting right due to inherent limitations. The neural approaches however, did have a chance of getting them right since the embedding inducing methods are capable of creating representations for the nodes involved by using other nodes which they are connected to besides the other node in the link in question. Thus the neural approaches would have information about these nodes beyond their immediate neighbourhood which it could use to make a decision.

For the MATADOR experiment, approximately 2% of the positive links had no prior common neighbours. Common Neighbours ranked none of these links in the top half of the rankings, but the best neural predictor ranked 26% there. In the BioGRID random-sliced experiment, approximately 16% of the positive links had no prior common neighbours. Common Neighbours ranked about 11% of these links in the top half, while the best neural predictor ranked 71% in the top half. For the time-sliced version, approximately 28% of the positive links had no prior common neighbours. Common Neighbours ranked about 21% of these links in the top half of the rankings, while the best neural predictor ranked 69% there. In the PubTator random-sliced experiment, approximately 2% of the positive links had no prior common neighbours. Common Neighbours ranked none of these links in the top half, while the best neural predictor ranked 51% there. For the time-sliced version, approximately 21% of the positive links had no prior common neighbours. Common Neighbours ranked about 11% of these links in the top half, while the best neural predictor ranked 57% there.

Note from those numbers that there was a marked increase in the number of links which formed which had no prior common neighbours in the time-sliced graphs. This relates to the point made in Section 4.6.1 about easy links forming chronologically earlier in a graph's evolution and underscores the need for evaluation of link prediction in time-sliced settings, especially when the aim is to aid LBD as this property applies to scientific knowledge.

Since the traditional ABC paradigm of open- and closed- LBD is inherently path based, this augments the arguments of the need to go beyond them to things like link prediction to really make interesting discoveries from scientific literature.

Issues with negative sampling method

As mentioned before, the negatives used for this experiment were randomly sampled from the set of nodes which had no links in the training data. This was a choice of convenience and there are potential issues with it. The main issue with this approach is that the majority of links which are created in this manner can be easy to spot as negatives because they will appear between nodes which are quite far apart in the graph. This is an additional explanation for the relatively high scores in the result tables, especially for precision at k. This issue is compounded by the fact that direction was ignored for these experiments so that a class of difficult negatives (nodes with a link in the reversed direction) was not available to evaluate the approaches on.

Summary

In general, for the neural network approaches, concatenate and average were the best node embedding combination techniques. Common Neighbours was the best baseline approach especially as graphs increased in size and remains quite an accurate heuristic for link prediction. In cases where the purpose of link prediction is to get only the very best links across the entire graph, then it almost does not matter which of the approaches is chosen for a small enough k, but if the quality of links at higher recall levels or the performance of the predictor across most nodes is essential, the choice of method is an important factor and the neural network approaches are clearly superior if they have enough data. For LBD the quality of links at higher recall levels or the performance of the predictor across most nodes should be considered essential.

The results showed that link prediction is a complex task which requires comprehensive experiments to determine best approaches, that performance is dependent on several things including the size of the graph and how it is split and that it is necessary to discern how a particular approach is achieving performance. It also highlighted that link prediction ought to be evaluated according to its intended purpose and that AUC metrics may not capture when and how well a particular approach works.

4.9 Link Prediction with Neural Networks Conclusion

In this work we investigated how node embeddings created with four graph embedding algorithms and combined with various methods perform on link prediction in biomedical graphs, with a neural link predictor. We tested in settings where links were randomly removed and where links are removed by time-slicing. We compared these methods to the performance of established baseline methods and reported performance on five metrics which aim to capture different facets of a link predictor's performance.

Our findings in both random- and time-sliced experiments indicate that where there is enough data for the neural network methods to learn good representations and there is a negligible amount of disconnected nodes, those approaches could perform much better than the baselines. However if the graph is small or there are large amounts of disconnected nodes, existing baselines such as Common Neighbours are a justifiable choice for link prediction. At low recall levels the approaches are basically equal, but at higher recall levels across all nodes and average performance at individual nodes, then the neural network approaches are clearly superior if they have enough data. We found evidence that the neural network methods do especially well in links which feature nodes with no previous common neighbours. We also found that while in general neural network methods benefit from large amounts of data, they require considerable amounts of computational resources to scale to large datasets. These findings provide large-scale comparisons and analyses that informs and explains the best approaches to link prediction and highlight areas of further development.

The neural network approaches to link prediction provide a truly promising way forward but they are not the best in all conditions and introduce added experimental considerations such as the creation of negatives and the combination of node representations. It is also well-known that the success of neural network methods greatly rely on hyperparameter tuning.

For future work it would be worth investigating the problem of creating good negatives for using machine learning methods for link prediction. Randomly creating negatives is experimentally valid but may create negatives which are not reflective of real-world difficulty. The problem of maintaining a large ratio of negative to positive links, as is the case in the real-world, without being computationally prohibitive is also worth exploring.

The models were developed with Python in Tensorflow. The Numpy, NetworkX (Hagberg et al., 2008), SciKit-learn, GEM (Goyal and Ferrara, 2018a) and Pandas (McKinney et al., 2010) libraries were also used. The code for the models used can be found at https://github.com/cambridgeltl/link-prediction_with_deep-learning.

4.10 Conclusion

General knowledge discovery, and link prediction specifically, makes sense as a logical method of performing traditional open and closed LBD and going beyond those paradigms. The advent of node embeddings and neural networks makes this possible using lots of existing tools. We investigated the feasibility of this in our work and obtained very promising results which indicated that neural network models which are given node embeddings from realistic biomedical graphs can perform very well at link prediction, especially in time-sliced settings for node-centric evaluations; two criteria which are important for LBD.

The approach of using node embeddings as input to neural networks which output some score of the probability of a path forming was shown to be successful here for links between two nodes. As LBD is a subset of link prediction, these results indicate that these techniques could be useful for generating high-quality suggestions from LBD systems. After the paper which introduced most of the work in this Chapter was published, a new LBD system claiming state-of-the-art performance was released (Pyysalo et al., 2018). It was developed with cancer researchers at two institutions and computer scientists working in collaboration. Instead of evaluating only on predicting links on graphs that were random- and time-sliced as we did here, they evaluated on five triples that represent specific recent discoveries (2011-2016) on the molecular biology of cancer that could have potentially been suggested by an LBD system in the past that were selected and curated by cancer biologists along with five pairs of Swanson's discoveries. Although these are a small amount of instances, this evaluation seems far more stringent and possibly more indicative of performance in the real-world.

The system used baseline methods that were similar to those used here and were mostly common neighbours-based. It was thus a logical next step to apply the methods presented in this Chapter to that system's data to measure their performance against a real-world system used by cancer researchers for scientific work and evaluated on published cancer discoveries. The next Chapter contains this work.

Chapter 5

Towards Integration – Comparison with a Real-world LBD System

5.1 Introduction

As stated in Section 2.3, the ultimate evaluation of an LBD improvement method or technique is its performance in the real world. At this point we stop just short of that high mark by evaluating methods and models for open and closed discovery, some inspired by those developed in Chapter 4, on real-world discoveries and compare their performance to a state-of-the-art, live system which was developed in conjunction with cancer researchers and also evaluated on the same discovery cases. We also applied them to a time-sliced dataset of human-curated, peer-reviewed biological interactions. These evaluations and the metrics they employ represent performance on real-world knowledge advances and are thus robust indicators of approach efficacy.

The relevant background on the system and the evaluation cases are presented, then details about the models and methods we developed were applied to the cases. We then compare my best performances to theirs, analysed the results and discuss their implications.

5.2 The LION LBD System

The LION LBD system (Pyysalo et al., 2018) enables researchers to navigate published cancer information and perform hypothesis generation and testing. It is focused on publications relating to the molecular biology of cancer processed using state-of-the-art ML and NLP methods, including NER and grounding to domain ontologies which include a wide range of entity types. It uses PubTator for annotating PubMed scientific articles with concepts

A	B	C	Reference
NF-κB	Bcl-2	Adenoma	Van Der Heijden et al. (2016)
NOTCH1	senescence	C/EBPβ	Hoare et al. (2016)
IL-17	p38α	MKP-1	Gaffen and McGeachy (2015)
Nrf2	ROS	pancreatic cancer	DeNicola et al. (2011)
CXCL12	senescence	thyroid cancer	Kim et al. (2017)
Migraine	-	Magnesium	Swanson (1988)
Somatomedin C	-	Arginine	Swanson (1990b)
Alzheimer's Disease	-	Estrogen	Smalheiser and Swanson (1996b)
Alzheimer's Disease	-	Indomethacin	Smalheiser and Swanson (1996a)
Schizophrenia	-	Calcium Independent Phospholipase A_2	Smalheiser and Swanson (1998)

Table 5.1 The Cancer Discovery and Swanson cases used to evaluate the LION System.

such as chemicals, genes/proteins, mutations, diseases and species; as well as sentence-level annotation of cancer hallmarks (Hanahan and Weinberg, 2000) that describe fundamental cancer process and behaviour (Baker et al., 2017a,b) according to the taxonomy of Baker et al. (2015). It uses co-occurrence metrics to rank relations between concepts and perform both open and closed discovery revealing indirect associations between entities in a database created from tens of millions of publications. An evaluation of the system demonstrates its ability to identify undiscovered links and rank relevant concepts highly among potential connections.

5.2.1 The LION Test Cases and Evaluation

These cases are described in detail in (Pyysalo et al., 2018). A condensed version is presented here for completeness.

To identify discoveries, the cancer researchers involved in the project first surveyed articles published between 2006 and 2016 in journals that publish works pertaining to biomolecular cancer, such as Science, Nature, The Lancet, British Journal of Cancer, and Cell. In the initial pass, they sought to identify specific cancer-related discoveries that can be characterized as a causal chain of three concepts, i.e. that fit the constraints of the traditional ABC paradigm of LBD. This initial literature survey yielded 50 candidate discoveries. The second stage filtered the candidates to identify discoveries that could have potentially been found by LBD: the two connections A-B and B-C should be found in the literature at some point in time before the connection between A and C is published. They identified cases where in some year in the past, A-B and B-C each co-occurred in at least 100 publications but where no or very few publications had A-C co-occur. To avoid possible bias towards a particular NLP methods or LBD tools the filtering was performed manually using PubMed

searches. In this manner the 50 candidates were culled to 16 which were then assessed by all project participants. This yielded a final set of 5 triples that represented specific recent discoveries on the molecular biology of cancer that could have potentially been suggested by an LBD system prior to their publication. The ontology and database identifier in the relevant resources were manually identified for each of the concepts in the dataset. In addition to these 5 cancer cases, in an effort to continue the trend of prior work, 5 cases from Swanson were also evaluated by the system. Details of these can be found in Table 5.1 which is adapted from (Pyysalo et al., 2018).

To evaluate LION using these cases, they used all combinations of metrics and scoring functions (explained in Section 5.2.2) and performed an open discovery query and a closed discovery query for each A-B-C triple using a version of the graph data that only includes literature up to five years before the year of the relevant publication (Table 5.1) and further excludes any document, regardless of publication date, where A and C co-occur. In open discovery, they query the system for nodes indirectly connected with the A node and determine the rank of the C node in the results. In closed discovery, they query the system for nodes connecting A and C and identify the rank of B. They summarise the results over the different test cases by reporting the average rank of the target node, using median as the average.

5.2.2 The Baseline Approaches

The baselines used for this part of the work are those used in the current version of LION LBD which claims state-of-the-art results. We present a condensed version here for completeness. The edge weight metrics which are currently implemented and a brief description of what they are follows (names in brackets are the shorthand they will be referred to going forward). Detailed definitions of these metrics can be found in the Supplementary Information of the LION paper.

- Co-occurrence count (Count): the number of sentences in which mentions of the entities connected by the edge co-occur.
- Document count (Doc-count): the number of documents in which mentions of the entities connected by the edge co-occur.
- Jaccard Index (Jaccard): the ratio of the size of the intersection over the size of the union of the sets of sentences in which the entities occur.
- Symmetric conditional probability (SCP): the product of the conditional probabilities of one entity being mentioned in a sentence where another occurs.

- normalized pointwise mutual information (NPMI): a measure of the independence of the mention occurrence distributions,
- Chi-squared (χ^2), Student's *t*-test (*t*-test) and log-likelihood ratio (LLR) are statistical tests measuring whether the mention distributions are independent of each other.

A number of alternatives for the scoring functions operating over the edge weights have also been implemented. For the aggregation function *f(g)*, the alternatives *min*, *avg*, and *max* are used. These functions assign the score for a path the minimum, mean, and maximum respectively of the edge weights on the path. For the accumulation function *f(c)*, the choices *sum* and *max* are supported. When multiple paths lead to the same node, the former sums the path score to obtain the node score while the latter simply uses the maximum score.

5.3 Models and Methods

This section contains the models and methods of my approach to the problem.

5.3.1 Evaluation

The post-cutoff years are used for evaluation. For the BioGRID dataset, this is randomly divided into development and test sets.

Cancer Case Discoveries

To facilitate direct comparison, we evaluate on the cases used in (Pyysalo et al., 2018), which describes them at length and for which a summary was provided in Section 5.2.1.

Time-slicing

The Cancer Discovery cases are strong evaluations for biomedical LBD systems as showing how a system would have ranked a discovery later published in a top-tier, peer-reviewed journal is a potent argument of its usefulness for LBD. However, the dataset is unsuitable for machine learning because it does not provide a development set to tune hyperparameters on; neither is it obvious how to create one. This meant that in the experiments with this dataset we had to evaluate its performance directly on the test cases which is not ideal for machine learning approaches. This, in addition to its limited size led us to seek additional evaluation methods to gain a more accurate picture of performance of our approaches and models.

For this we chose a dataset which contained human-curated biomedical interactions which were published in peer-reviewed journals (details in Section 5.3.4). A graph created from the interactions in this dataset is time-sliced. From the post cut-off publication year, development and test sets are constructed. In some senses, this is not as stringent an evaluation and it is not possible to do closed discovery with it, but this provides robust additional evaluation of our open discovery approaches on a larger test set which is more indicative of approach generalizability.

Metrics

The evaluation metrics used are important when analysing the performance of ranking systems. Pyysalo et al. (2018) reported median ranks over the groups of cases for the case discoveries. For comparability, we shall also report this along with the mean over the cancer and Swanson cases separately and combined.

For the time-sliced experiments, we will also report MAP, Mean Reciprocal Rank (MRR) and Mean R-precision. There are 2 reasons for this: there is great variance between the amount of Cs which are ranked for each A so the mean rank can vary widely and distort the results; and these metrics, especially the latter 2, give higher priority to correct scores ranked highly in the list. This is of importance in any ranking problem but especially so for LBD where investigating each proposal is a costly endeavour. Formal definitions of these evaluations are in Appendix C.

5.3.2 Baselines

The baseline approaches are those used by Pyysalo et al. (2018). They are 8 co-occurrence metrics accompanied by 3 aggregator functions and 2 accumulator functions (for open discovery). Details can be found in the referred paper: Section 3.3 and full details in the Supplementary Information. We focus on only the best performing methods for the mean (and standard deviation) and median metrics and report the relevant accumulator and aggregator functions in each experiment.

5.3.3 Neural Approaches

Two neural link prediction models and methods are used for closed discovery and another two for open discovery. All approaches use node embeddings created with LINE with weighted edges, where weights are calculated using Jaccard Index. The embeddings were induced with the portion of the graph used for training, the pre-cutoff year period. The settings used are in Appendix C.

For each of the approaches described here, the same five node combination methods as in Section 4.6.1 and defined in Table 4.1 were used to determine how the nodes which constitute the link path were combined for input into the model. Here, models ending in '-A' refer to approaches which used Average to do this, '-C' - Concatenation, '-H' - Hadamard, '-W1'- Weighted-L1 and '-W2'- Weighted-L2.

Closed Discovery neural models and approaches

In both of these approaches the model was a Multi-Layer Perceptron (MLP) which was effective in the similar task of neural link prediction on biomedical graphs (Chapter 4). The model contains a single hidden layer with ReLU activation which led to a final layer with Softplus activation to allow for unrestricted positive scores. The model was trained as a classifier with the Cross Entropy loss.

CD-1: The neural model is used to provide a score for each A-B and B-C link in the path. The scores are then used in aggregator functions as the baseline methods, so the neural network in effect replaced the metric calculation.

CD-2: In this approach A-B-C vectors are combined to create a single input to the model which predicts a score for the entire A-B-C link. This negates the need for an aggregator function as in the baselines and CD-1 approach and allows the approach to be (mostly) indifferent to the length of the path between A and C.

Open Discovery neural models and approaches

OD-1: The same model and a similar approach to CD-1 was used here. The difference was that here the scores are then used in the aggregator and accumulator functions which the baseline methods use.

OD-2: A CNN was used to implement an approach to open discovery which removes the need for aggregator and accumulator functions. In this approach, the A-B-C path for each A-C link constitutes a window which we pass into the CNN which outputs a score indicative of the strength of the A-C links. This is analogous to applying a CNN over images but here the 'image' is produced by stacking combined vector representations of ABC link. The convolutional filter always slides down the stack of links, never across so that it always covers the entire link. The ABC links to be stacked are combined using the same 5 link combination methods mentioned above. The CNN expects a fixed size input and the amount of intermediate connections vary from case to case, so the links were combined into a fixed window size using elementwise summation. Zero padding was used to fill any remaining gaps in the window.

In this model, the input layer led to a batchnormed convolutional layer with ReLU activation units, then a max pooling layer then a fully connected layer before the final layer with Softplus activation. Unlike the other models which are trained as classifiers, this model uses a pointwise approach, employing Mean Squared Error (MSE) loss, to learning the ranking function by using the Jaccard Index score of the AC link as the multi-level ratings (see Chen et al. (2009)).

5.3.4 Datasets

The graphs used were created from the following datasets. The graph details can be found in Table 5.2.

PubTator:

Biomedical entities recognised by PubTator (Wei et al., 2012, 2013) mentioned in the titles and abstracts of PubMed publications from 1873 to 2017 were used to create this dataset. A link exists between two biomedical entities if they co-occur in a single sentence. The annotations were downloaded on June 20th, 2017. Instances of Hallmarks of Cancer identified in text are also featured in this graph.

Biological General Repository for Interaction Datasets (BioGRID):

This is an open database created from manually curating experimentally-validated genetic and protein interactions that are reported in peer-reviewed publications (Stark et al., 2006). The latest major release (Chatr-aryamontri et al., 2017) includes over 1 million Genetic and Protein interactions across all major organism species and humans. Links in this graph represent biomedical interactions from published, experimentally-validated genetic and protein interactions. We use version 3.4.167 of this dataset.

Dataset	Node Count	Link Count	Link Type
BioGRID	68,734	1,209,578	Published Interactions
PubTator	$\sim$194,691	$\sim$12,797,468	Literature Co-occurrences

Table 5.2 Graph details (undirected link count)

5.4 Experimental Settings

As all approaches create ranked lists, the possibility of tied ranks exists. We use the median of the tied range to determine the rank of a gold item with ties, for example a gold ranked 10th with 10 ties is ranked the median of 10-20 range: 15th.

5.4.1 Details of Neural Approaches

Unlike the baseline models, the neural approaches need negative examples for training. We created these by selecting either A-B or B-C links which did not form for a given A-C or A-C connections which did not exist for models which operated on the entire link path (i.e. those without accumulators or aggregators).

All models are trained with batch size 100, training set size 200,000 for 150 epochs with the Adam optimiser (Kingma and Ba, 2015), but the model is evaluated on the case after every 5 epochs and the best performance reported. For the BioGRID experiments, because evaluation is a lot more time-consuming, the models are evaluated every 25 epochs on the development set and the best performing model on MRR is evaluated on the held out test at the end. The CNN uses a learning rate of 10^{-5} while the MLPs use 10^{-4}. For CD-1, CD-2 and OD-1, there is a single hidden layer with 100 units. For OD-2, the input height is 50 and the width is the size of the combined vector dimensions. The convolution window height is 7 and the convolutional output size is 128.

5.4.2 Case Discoveries

We used the data from Pyysalo et al. (2018) directly, so that our results will be directly comparable. The graphs were cut off at the relevant years before the publication date of the discovery as mentioned above.

Cancer Discoveries Closed Discovery: For CD-1, the model was fed the A-B and B-C links and the scores it produced were used in the aggregator functions to rank the Bs. For CD-2 the model was fed all the A-B-C links for the given A and C in each triplet and the score it produced was used to rank the Bs.

Cancer and Swanson Discoveries Open Discovery: For OD-1, the model was fed the A-B and B-C links and the scores it produced for each link were used in the aggregator functions to produce a score for each path. The different paths which led to the same C were used in the accumulator functions to produce a score used to rank the Cs. For CD-2, the model was fed all the A-B-C links for the given A and C in each pair and the score it output was used to rank the Cs.

5.4.3 BioGRID

The graph is split at the year 2016. We randomly split the post-2016 links into development and test sections. The development set is used to determine which epoch has the best trained model for evaluation. Due to computational constraints, we have to restrict the amount of nodes we could evaluate on. We randomly select 1,000 entities from the test set to be A nodes and have the model score each node within two hops as the Cs. The scores are then used to rank the Cs. Like the Swanson cases, it is not possible to perform closed discovery on this dataset.

5.5 Results

The results of the neural approaches are means of the means and medians which were calculated over 5 runs. The standard deviations reported are of the mean ranks. The results of the baselines are means of the method across all relevant cases and the standard deviations are those over those ranks. The best score for a metric is in **bold** and the best for an approach is underlined; all the baselines methods are treated as a single approach. For the cases, we sought to determine what methods gave the lowest mean and median ranks and lowest variance (measured by standard deviation). For BioGRID experiments, we sought the lowest MR, but the highest MRR, MAP and R-precision. To increase clarity in the tables, only the best results for each approach was selected to be shown here. Full experimental results for all approaches in all experiments can be found in Appendix C.

5.5.1 Closed Discovery: Cancer Discovery Cases

The results for closed discovery performed on the five Cancer discovery cases used to evaluate LION are in Table 5.3.

Approach	Mean Rank	Std. Dev.	Median	Details
Jaccard	214.8	256.9	81.0	Agg: min
***t*-test**	262.0	341.8	56.0	Agg: min
CD-1-A	86.3	52.0	93.8	Agg: min
CD-1-C	94.5	80.0	**36.4**	Agg: min
CD-2-C	**48.7**	19.5	42.0	-

Table 5.3 Closed discovery: Mean and Median ranks on the Cancer Discovery cases

5.5.2 Open Discovery: Cancer Discovery and Swanson Cases

Open discovery on only Cancer Cases

The results for open discovery performed on the 5 Cancer Discovery cases used to evaluate LION are in Table 5.4.

Approach	Mean Rank	Std. Dev.	Median	Details
NPMI	60.2	54.4	36.0	Acc: sum, Agg: max
Count	367.4	553.3	15.0	Acc: sum, Agg: min
OD-1-C	93.4	145.8	31.4	Acc: sum, Agg: min
OD-1-A	218.3	368.7	26.8	Acc: sum, Agg: min
OD-2-H	**31.1**	11.9	**12.2**	-

Table 5.4 Open discovery: Mean and Median ranks on the Cancer Discovery cases

Open discovery: Swanson Cases

The results for open discovery performed on the five Swanson cases used to evaluate LION are in Table 5.5.

Approach	Mean Rank	Std. Dev.	Median	Details
Doc-Count	2,199.8	4,216.7	31.0	Acc: max, Agg: avg
***t*-test**	3,956.4	7,899.3	5.0	Acc: max, Agg: avg
OD-1-H	3,558.3	7,930.7	19.2	Acc: sum, Agg: min
OD-1-C	3,721.4	8,306.7	**4.0**	Acc: sum, Agg: min
OD-2-H	**1,013.4**	167.9	17.6	-

Table 5.5 Open discovery: Mean and Median ranks on the Swanson Cases

Open discovery: Cancer Discovery and Swanson Cases

The results for open discovery performed across the five Cancer Discoveries and five Swanson cases combined are in Table 5.6.

5.5.3 Open Discovery: BioGRID Published Interactions

Results for open discovery performed on the BioGRID dataset. Performance across the 4 metrics explained in Section 5.3.1 are in Table 5.7.

Approach	Mean Rank	Std. Dev.	Median	Details
Jaccard	1,634.4	4733.9	21.0	Acc: sum, Agg: min
Count	1,925.8	5171.3	**11.5**	Acc: sum, Agg: min
OD-1-C	1,907.4	5,859.4	18.2	Acc: sum, Agg: min
OD-2-H	**522.2**	89.9	14.9	-

Table 5.6 Open discovery: Mean and Median ranks on all open discovery Cases

Approach	MR	MRR	R- Prec.	MAP	Details
Jaccard	1,197.3	2.19	2.47	2.86	Acc: sum, Agg: min
LLR	1,132.9	1.34	1.38	1.9	Acc: sum, Agg: max
OD-1-H	1,907.5	0.92	0.96	1.25	Acc: sum, Agg: max
OD-1-C	1,913.4	0.94	1.01	1.23	Acc: sum, Agg: max
OD-1-W2	1,908.3	0.92	0.98	1.26	Acc: sum, Agg: max
OD-2-C	**1,113.1**	**3.42**	**4.73**	**5.46**	-

Table 5.7 Open discovery on time-sliced BioGRID

5.6 Discussion

5.6.1 Cancer Discovery and Swanson Cases

Closed discovery on Cancer Discovery cases

The neural approaches performed much better than the existing methods in these experiments. The performance measured by mean ranks doubled by simply replacing the metrics with a small neural classifier to provide the scores instead. It almost doubled again by replacing the aggregation of individual path scores with combining the vectors of the nodes involved in the path. Performance on the median also increased though not as drastically.

Of note here is that the neural approach which dispelled with the aggregator functions, instead opting to combine the inputs and obtaining a score for the entire path, was the best performer on mean ranks and the second best performer on median. This indicates that the information which the aggregator functions seek to provide to an approach is better provided by combining the vector representations of the nodes in the path.

Open discovery on Cancer Discovery cases

Despite the strong improvements seen in closed discovery by simply replacing the scoring metrics with a neural classifier, that was not the case here in both mean and median ranks. However, the more complex CNN approach was able to produce results which approximately

doubled performance from a strong baseline. It was also able to perform the best on median ranks.

Analogous to the closed discovery experiments, the approach which dispelled with aggregators and accumulators outperformed on mean ranks. Additionally, it was the best median performer here, further validating it.

Open discovery on Swanson cases

A similar trend to the cancer cases was shown here: simply replacing the metrics with a neural classifier decreased performance on mean rank, although one such approach did produce the best median rank. The strong performance of the CNN continued as it again doubled performance on mean rank although it was only the third best on median rank. The trend of the approach which dispelled with aggregators and accumulators outperforming on mean ranks also continued.

Open discovery on both Cancer Discovery and Swanson cases

Given the results of the subset experiments, it is not surprising that the CNN was the best performer across all open discovery cases. Its performance on mean rank was approximately three times better than that of the best baseline and it was the second best on median, although the simple count baseline approach was the best.

General open discovery

In addition to its strong performance across the cases, the OD-2-H approach is also quite stable as it showed the lowest variation in performance over multiple runs of the best performing methods.

A point in favour of the neural approaches presented here over the baselines is their apparent consistency in performance over the subsets of the cancer and Swanson discoveries. The baseline methods which performed the best over Tables 5.4, 5.5 and 5.6 varied while the best neural approaches recurred, demonstrating their invariability to the vagaries of the subsets of the cases.

General case discoveries

Whether to use mean or median as average for these experiments is a valid question. Pyysalo et al. (2018) reported median and we do the same here to allow for comparison while reporting the mean as well because we believe that it is better adapted to this situation. The median is

robust to outliers and can give a more accurate picture of an approach's performance when an outlier can radically affect the mean as is the case with the Swanson cases used. However, the aim of this research is to find an approach which will aid researchers on unseen data, so the worst-case performance of the system (even if it is rare) is of importance and the aim should be to use methods which will give the best results across all cases. Thus, evaluating accurately should involve looking at performance in all available cases and median ignores not only outliers, but effectively all performances which are beyond the median ($\sim$ half of the use cases). The argument can thus be made that the median does not give a true reflection of an approach's performance.

Taking mean as a preferable metric to median for this situation, the case of the neural methods is strengthened as they were the best performers across all the case experiments. Additionally, there was low variance among the best neural approaches. It was also pleasing to find that approaches which dispelled with the cumbersome aggregator and accumulator functions were the best. This highlights that when given the full path information, the neural models are able to discern how best to use it to improve performance.

It is also worth noting that although methods which concatenated the node representations performed well, there were other approaches whose performance were comparable or better than it across these experiments. This is of significance because the other node combination methods are indifferent to the amount of hops between A and C, which makes them amenable to approaches to LBD beyond the simple two-hop ABC paradigm which it is generally agreed must be overcome for LBD to reach its true potential.

5.6.2 Time-sliced BioGRID

The reasons for undertaking these experiments were explained in Section 5.3.1 and the reasons for the multi-faceted evaluation in Section 5.3.1. We will make use of and expand on these here.

The graph used in this experiment represent experimentally validated, human-curated interactions which were published in peer-reviewed publications. Thus the knowledge proposed by the approach is of high quality. Additionally, the evaluation is a time-sliced one which is reflective of how knowledge discovery progresses in the real world, is a more difficult type of evaluation and involves far more evaluation instances than a handful of cases, notwithstanding the very high quality of the cases.

LBD across a large amount of possible positives is a ranking problem because its proposals are usually costly to investigate. Thus priority should be given to approaches which can rank correct new associations at the very top of the list even if they rank more of them lower; the classic precision-recall trade-off. Performance too far down the list can effectively

be ignored: when experimentally validating new knowledge proposals, whether it is ranked 200^{th} or 900^{th} is likely of little concern to a user; it is too far down the list.

Metrics like MAP, MRR and R-precision place value on higher ranked true positives but they do not do so equally. MAP and MRR are concerned with the entire list but MRR punishes lower-ranked correct items more when the retrieval space is large as it tends to be in LBD, especially open discovery. R-precision literally discards most of the returned results and reports results only on the best. Thus performance on metrics like R-precision and MRR give a better idea of the practical worth of an LBD system, especially on open discovery.

The OD-2-C method we introduce here performs approximately 1.5-1.9 times as good the baseline approaches on these metrics, in addition to strong performance on MAP and mean rank. It is a minor variant of the OD-2-H method which showed vastly better performance on the cases experiments. The results here thus validates the OD-2 (CNN) approach to open discovery we presented in Section 5.3.3. It is a shortcoming of this evaluation that closed discovery couldn't be performed here, but based on these results, there is an indication that neural network approaches without aggregators would have performed the best.

While there is still lots of room for improvement, these results are dependable and demonstrate the potential for using neural networks to perform even traditional open and closed discovery within the ABC paradigm.

5.7 Conclusion

The ultimate evaluation of an LBD technique or improvement method is its performance in the real world. At this point we stop just short of that by presenting and evaluating four methods and models for open and closed discovery, some inspired by those developed in Chapter 4, on real-world discoveries and compare their performance to methods used in a state-of-the-art, live system which was developed in conjunction with cancer researchers and also evaluated on the same discovery cases. We also applied them to a time-sliced dataset of human-curated, peer-reviewed biological interactions. These evaluations and the metrics they employ represent performance on real-world knowledge advances and are thus robust indicators of approach efficacy. In both cases, our methods showed a notable and significant improvement over the existing methods on metrics adapted to the situation.

Although there is scope for much improvement, these results demonstrate the strong potential of using neural networks to perform open and closed LBD well, even within the flawed and maligned ABC paradigm and in some cases using the inferior co-occurrence relationship. Combined with the work of Chapter 4 on the viability of using neural link

prediction for LBD, it seems clear that neural networks can significantly improve performance on this increasingly important task.

The models were developed with Python in Pytorch (Paszke et al., 2017). The Numpy, NetworkX, SciKit-learn, and Pandas libraries were also used. The code for the models used can be found at https://github.com/cambridgeltl/nn_for_LBD.

Chapter 6

Conclusion

6.1 Introduction

In this Chapter we recap the motivations and issues which instigated this research, provide a synopsis of the work done and highlight the salient findings of this research along with their implications. We revisit the potential of LBD for use in modern biomedical research and the technical problems it still has and how they can be solved. We also deal with how the work proposed in this thesis can be solutions to those problems, main findings from our investigations and the implications of those findings for LBD. We then close off by looking at possible directions in which this work can be taken in the near future.

LBD seeks to discover new knowledge from existing literature in an automated or semi-automated way. The biomedical domain has been its main test-bed so far because of its potential for great impact there. Since LBD generates new knowledge by combining existing literature, the possibility of using computers and algorithms to discover biomedical connections automatically in large collections of literature is tantalising. It can potentially facilitate both testing and generation of complex hypotheses from such collections of literature and support or accelerate scientific research.

Scientific literature is growing exponentially, making it difficult for researchers to stay current in their discipline. This overwhelming volume of publications and the increasing need to specialise has led to the creation of non-interacting literature silos, which engenders an environment where discoveries in one area are not known outside of it and valuable logical connections between disparate bodies of knowledge remain unnoticed. In such an environment, there is a very real chance that slivers of information which can be combined to make breakthroughs are already discovered but are dispersed throughout the literature. LBD can solve these problems by combining these slivers to help researchers quickly gain

information on relevant advances inside and outside of their respective niches. As the scientific literature grows, LBD is increasingly becoming a necessary research tool.

6.2 Research Motivation and Synopsis

Despite the promising applications found for LBD thus far and its potential for knowledge discovery and increased research efficiency, at present LBD systems are yet to see widespread adoption and any meaningful uptake by those who can potentially benefit the most from them. There are several reasons for this, some of which are non-technical, but there are also technical shortcomings in existing LBD approaches which negatively impact their performance and hinder their application in real-world environments.

These include producing an over-abundance of low-quality discoveries; high dependence on static external resources for entity and relationship recognition in literature; simplistic approaches to expressing biomedical relationships such as co-occurrences; and an over-reliance on the restricted ABC paradigm which was proposed at the field's conception. All of this is exacerbated by the lack of comprehensive evaluation methods and metrics which would allow direct comparisons and analyses of the merits of a proposed approach or improvement. We propose that the relevant computational technologies are at the stage where LBD systems can process literature to extract real information from it and produce viable high-quality discoveries to result in usable output. We also make use of robust, reusable evaluations of our approaches which allow quantification of their performance and comparability with other approaches.

This work presented research to neurify LBD. It used neural networks to improve the performance of a task which produces input for LBD; applies them to an LBD approach which can also surpass LBD as an avenue for knowledge discovery from biomedical literature; and on increased performance on the current dominant ABC paradigm while using methods which can circumvent some of its weaknesses. Some of the methods presented here were applied to evaluations used in a real-world, state-of-the-art LBD system.

There has been an abundance of work on LBD since it was proposed as a strategy for discovering new scientific knowledge. These works have proffered several different approaches to LBD even when they used some similar components. The evaluation methods used have been more prosaic: there is heavy use of seeking to replicate specific discoveries especially Swanson's although there have been other approaches to evaluation as well. Unfortunately, in general very few evaluations can lay claim to demonstrating that an approach is generalisable and practically useful. There have also been several approaches to improving LBD either through improving how it extracts useful information from text for use

in LBD or in performing LBD more efficiently and obtaining high-quality proposals from the LBD process itself. The existing methods mostly use very shallow processes to extract the entities in text by matching to a list of known terms from lists created by external resources. That makes these methods reliant on these resources which are generally incomplete and lag the current state of knowledge. It additionally makes them error-prone as simple string matching tends to produce false positives and false negatives.

NLP presents a solution to entity recognition in text and while there will be a trade-offs, the benefits would outweigh this downside; we explored this in Chapter 3. Link prediction provides both an alternative approach to facilitating simple (single-hop) and multi-hop LBD and a more powerful approach to knowledge discovery from biomedical knowledge represented as graphs than the traditional open and closed ABC paradigm of LBD; we explored this in Chapters 4 and 5. Advances in neural networks and deep learning have made applying all these techniques to improving LBD possible, feasible and highly promising given their stellar performances on other tasks.

The overarching goal of the PhD was to investigate possible areas of research to improve the performance of the tasks which produce input for LBD as well as LBD itself. Specifically, the task which produce input for LBD was biomedical NER, link prediction on biomedical graphs was the LBD approach and we also applied neural networks to the ubiquitous ABC paradigm, evaluated on published discoveries. They all relied on recent advances in neural networks in several areas including improved development and training of deep neural models and enhanced word and graph representations.

6.3 Contributions: Work Completed and Important Findings

6.3.1 Neural Biomedical NER

NER is an important precursor to LBD and knowledge discovery. Various advances in neural networks have opened new avenues for improved NER. These include semantically-rich word representations (embeddings), MTL with CNNs and character-level features with LSTMs. Leveraging these methods, we engaged in work on three aspects of NER with the aim of improving its performance for biomedical NER. We worked on extrinsic evaluation of improved biomedical word embeddings as inputs to neural NER models, MTL with CNNs to harness the information in disjoint datasets and on attention-based, character-level sequence labelling for NER.

The word embeddings developed were evaluated using a CNN model we developed for NER on two biomedical NER datasets. When trained with the *window size* hyperparameter set to 1, they led to improved performance over the existing embeddings for biomedical NER.

For investigations into MTL, we developed a single-task CNN model and then two variants of a multi-task CNN. We trained these on several datasets representing a wide range of biomedical named entities. We observed an average improvement from the MTL models in comparison with single task learning. Individually, there were also significant improvements on many of the datasets. There was a drop in performance on some tasks, but for most tasks performance improves significantly. We also found that MTL is beneficial for small datasets. Across the various settings the improvements are significant, demonstrating the benefit of MTL for biomedical NER.

The investigations into character-level extensions to models for sequence labelling tasks such as NER led to an architecture for combining alternative word representations. By using an attention mechanism, the model is able to dynamically decide how much information to use from a word- or character-level. We evaluated different architectures on a range of sequence labelling datasets, and character-level extensions were found to improve performance on every benchmark, especially the biomedical NER datasets. The proposed attention-based architecture delivered the best results.

6.3.2 Neural Link Prediction

In this work we investigated how node embeddings created with four graph embedding algorithms and combined with various methods perform on link prediction in biomedical graphs with a neural link predictor. We tested in settings where links were randomly removed and where links are removed by time-slicing. We compared these methods to the performance of established baseline methods and reported performance on five metrics which captured different facets of a link predictor's performance.

The findings in both experiments indicate that where there is enough data for the neural network methods to learn good representations and there is a negligible amount of disconnected nodes, those approaches could perform much better than the baselines. However, if the graph is small or there are large amounts of disconnected nodes, existing baselines are a justifiable choice for link prediction. At low recall levels the approaches are basically equal, but at higher recall levels across all nodes and average performance at individual nodes, then the neural network approaches are clearly superior if they have enough data. We found evidence that the neural network methods do especially well on links which feature nodes with no previous common neighbours.

The neural network approaches to link prediction provide a truly promising way forward but they are not the best in all conditions and introduce added experimental considerations such as the creation of negatives and the combination of node representations. General knowledge discovery, specifically link prediction makes sense as a logical method of performing traditional open and closed LBD and going beyond those paradigms. The advent of node embeddings and neural networks makes this possible using lots of existing tools. We investigated the feasibility of this and obtained very promising results which indicated that neural network models which are given node embeddings from realistic biomedical graphs can perform very well at link prediction, especially in time-sliced settings for node-centric evaluations - which are important for LBD.

6.3.3 Neural LBD

The ultimate evaluation of an LBD technique or improvement method is its performance in the real world. We stopped just short of that by presenting and evaluating four methods and models for open and closed discovery, some inspired by those developed for link prediction, on real-world discoveries and compared their performance to methods used in a state-of-the-art, live system which was developed in conjunction with cancer researchers and also evaluated on the same discovery cases. For additional evaluation, we also applied them to a time-sliced dataset of human-curated, peer-reviewed biological interactions. These evaluations and the metrics they employ represent performance on real-world knowledge advances and are thus robust indicators of LBD approach efficacy. In both cases, our methods showed a notable and significant improvement over the existing methods on metrics adapted to the situation.

Although there is scope for much improvement, these results demonstrate the strong potential of using neural networks to perform open and closed LBD well, even within the flawed ABC paradigm and in some cases using the simplistic co-occurrence relationship. Combined with the work on the viability of using neural link prediction for LBD, it seems clear that the inexorable spread of neural networks will arrive at the increasingly important task of LBD where it can significantly improve performance.

6.4 Implications of Findings for LBD

6.4.1 Neural Biomedical NER

This work indicates that it is possible to create a biomedical NER model which can harness multiple annotated datasets to perform well at recognising various biomedical entities in

unstructured text. It also showed the benefits of MTL for small datasets. There are two main implications of this work.

The first is that the multiple NER datasets in existence for a cross-section of biomedical entities can be utilised in a way which may create a model which can perform biomedical NER at a high level. It is possible that this could lead to improved recognition and extraction of biomedical entities in unstructured text which will lead to better inputs to an LBD approach and thus improved LBD. Such a model will also be expected to deal with new biomedical entities introduced in the literature with little or no re-training, diminishing the negative effects of limited and outdated vocabulary and error-prone term, concept or keyword matching.

The second is that for entities for which there are only small datasets, neural approaches can be applied to extract them well anyway by training a multi-task model with larger datasets which already exist. An implication of this is also that when it becomes necessary to create a new dataset for a new class of biomedical entities, the dataset can be small. This will allow it to be completed quicker and with less human and financial resources.

The work on improving biomedical word embeddings for NER shows that it makes sense to train specialised embeddings for the biomedical domain and to adjust the hyperparameters to create embeddings which are better suited to the task which the embeddings will be used. The character-level LSTM approach to NER showed that it is beneficial to incorporate character-level information into models designed for biomedical NER and using methods like character-level attention could further enhance performance. This is easily justified due to the amount of information that biomedical entities encode at the character-level.

6.4.2 Neural Link Prediction

This work showed that it was possible to perform neural link prediction in large-scale biomedical graphs in realistic settings like time-slicing. The methods here showed improved performance as the amount of data to induce embeddings and train models increased. The amount of high-quality biomedical data represented as graphs are plentiful, and any method capable of exploiting their network structure to improve performance on tasks which use them is welcome. That the neural methods were also the better performers on time-sliced graphs and on node-centric evaluations also has positive implications since time-sliced approaches closely mimic the progress of biomedical knowledge evolution and LBD is an inherently node-centric task.

It also showed that the neural methods were able to perform best on links involving nodes which had no common neighbours. In traditional LBD parlance, this means discoveries with no close linking terms. This highlights the approach's ability to progress past the simple ABC

paradigm popularised in the field's nascent stages and which it is now commonly agreed must be superseded for LBD to reach its true potential. Such discoveries are also likely to be the most unexpected and thus interesting ones; the type which LBD systems are expected to provide, since humans are unable to without serendipitous events.

6.4.3 Neural LBD

The implications for the results of this work are easier to see than the others. The obvious implication is that once these methods are implemented in the LION LBD system, its performance will improve. This makes it conceivable that a useable system with the ability to test and generate hypotheses which is freely-available to all cancer researchers with an internet connection will return improved results which could positively impact their work - bringing it one step closer to the dream of an LBD system which can provide concrete support to scientific research.

Another feature of this work is that it used graphs of simple sentence-level term co-occurrences. Its results indicate that the approaches proposed were able to still make use of the noisy input to produce good quality results. For all the arguments against it, co-occurrence continues to be used because of its simplicity and scalability; if there are approaches which can filter out co-occurrence noise with large enough datasets then they will be embraced. In a similar vein, it also used the simple ABC paradigm to perform well on the evaluations used; illustrating that although it is flawed, a powerful enough model will be able to still glean useful signals to perform well within it. Note that these are not arguments against using more intelligent relationships and paradigms, but rather suggest positive features of the proposed models and hints that performance may improve further with more intelligent approaches.

Another advantage of this work is that although it was evaluated on the ABC paradigm, like the link prediction methods, it is also capable of going beyond it. All the node combination methods except concatenate are indifferent to the length of the path between A and C. The only open question is how each will be affected by the repeated vector arithmetic as the path length grows. However, given the diversity in the way in which they approach the node combination, it is conceivable that at least one will still perform well on longer paths which illustrate the $A - B_1 - B_2 - ... - B_n - C$ paradigm of open discovery.

6.4.4 General

This work set out to investigate the feasibility of using neural networks to improve LBD in general. Its results showed that it is in fact feasible to use neural networks to improve LBD at different points in the pipeline. It also showed that neural networks are versatile

enough to be applied to improve traditional approaches to LBD as well as different forms of non-traditional LBD including LBD with longer path hops between starting term and discovery candidate and even no path hops. The principal implication of the findings of this work is that neural knowledge discovery, especially LBD is potent and ready for present use in addition to being a potentially rich field for further study.

6.5 Future Work and Directions

In earlier chapters, we highlighted several shortcomings of the current state of LBD and possible ways they can be rectified. We also mentioned several ways in which neural networks can be used for LBD. Some of the shortcomings were not addressed in this work and some of the ways which neural networks can be applied to LBD were not investigated. These remain prospective research areas and we catalogue the most prominent here, although the list is longer.

6.5.1 Relation Extraction

LBD requires concepts and the relationships between the concepts to work. Work was done on improving the extraction of concepts from free text, but not the relationships. Neural relation extraction has already received some attention in both the biomedical domain (Barnickel et al., 2009) and the general domain (Lin et al., 2016; Nguyen and Grishman, 2015; Zeng et al., 2015).

Of interest in this area is also a recent machine learning approach to predicting Adverse Drug Events using embedded relations (Mower et al., 2016). Once these relations are embedded and are represented with a vector of real numbers then they can be seamlessly integrated into our neural approaches. Works in the general domain already use relations embeddings (Lin et al., 2015; Yang et al., 2014).

6.5.2 Integrating Information in Knowledgebases

We mentioned the wealth of information available in biomedical knowledgebases (detailed in Section 2.6.1) but they are never harnessed explicitly in any of the work presented here. On the face of it, this seems easy to do as they are mostly already formulated as graphs so the graph embedding algorithms used could perhaps be applied to them easily. The resulting vectors could then be combined with those of the vectors created from the graphs of the datasets used to obtain a final representation which contains information from multiple

sources. This was mostly left undone due to the difficulty in reliably mapping the name of an entity in one dataset to its equivalent name in another dataset.

6.5.3 Integrating Information from Non-Literature Sources

Biomedical researchers use a range of tools and sources besides published papers to derive their understanding and create hypotheses. These include tools which map pathways, among others. LBD methods should also make use of these resources and the information within them to realistically aid researchers in understanding the state of their field and suggest possible advancements. One example of how this may be done was mentioned in Section 2.4 where Hristovski et al. (2003) used knowledge of chromosomal locations of diseases and genes from LocusLink and OMIM to require that candidate genes be in the same location as the diseases they are involved with.

6.5.4 Using Improved Graph Embedding Methods

The graph embedding methods used in this work produced results which advanced the state of the field on the tasks they were applied to. However, they were the first generation of works which used neural approaches for embedding graphs. Since their publication, there has been an explosion of other methods which claim to be significantly better than they are. Of note are graph convolutional networks (Kipf and Welling, 2017) and GraphSAGE (Hamilton et al., 2017). The hope is that improved embeddings would then lead to improved graph-based methods to perform LBD.

6.5.5 End-to-end Neural LBD

This is not so much an unexplored avenue of this work as it is a newly feasible, intriguing next level. Neural networks have gained a reputation of performing well when they are used for end-to-end task and LBD could be one such task. End-to-end models are models which replace a pipeline which may or may not include neural networks and feed the original input to the neural network and collect the final outcome at the end, letting the neural network perform the steps that were previously separated in the pipeline together in its hidden layers. This usually results in more efficient performing of tasks and avoids propagating errors made earlier in the pipeline.

For LBD this would translate to passing the embeddings of the text to such a model and having it output the new discoveries without explicitly performing concept or relationship extraction, or the actual discovery methods. In addition to the increased performance seen in

other domains on this, we also saw a micro version of this in Chapter 5 when the methods which dispelled with aggregators and accumulators were able to outperform the methods which did. In that approach, those manual steps (and their limitations and biases) were replaced with a model that took the entire path and found its own way of using the condensed input it was given to perform better at the task. The main disadvantage of such a model would be that it would be difficult to interpret how it arrived at its results and to debug it.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.

Adamic, L. A. and Adar, E. (2003). Friends and neighbors on the web. *Social networks*, 25(3):211–230.

Ahlers, C. B., Hristovski, D., Kilicoglu, H., and Rindflesch, T. C. (2007). Using the literature-based discovery paradigm to investigate drug mechanisms. In *AMIA Annual Symposium Proceedings*, volume 2007, page 6. American Medical Informatics Association.

Al Hasan, M., Chaoji, V., Salem, S., and Zaki, M. (2006). Link prediction using supervised learning. In *SDM06: Workshop on link analysis, counter-terrorism and security*.

Alonso, H. M. and Plank, B. (2017). When is multitask learning effective? semantic sequence prediction under varying data conditions. In *EACL 2017-15th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–10.

Ando, R. K. and Zhang, T. (2005). A framework for learning predictive structures from multiple tasks and unlabeled data. *J. Mach. Learn. Res.*, 6:1817–1853.

Apweiler, R., Bairoch, A., Wu, C. H., Barker, W. C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M., et al. (2004). Uniprot: the universal protein knowledgebase. *Nucleic Acids Research*, 32(suppl_1):D115–D119.

Argyriou, A., Evgeniou, T., and Pontil, M. (2007). Multi-task feature learning. In Schölkopf, P. B., Platt, J. C., and Hoffman, T., editors, *Advances in Neural Information Processing Systems 19*, pages 41–48, Cambridge, MA. MIT Press.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., et al. (2000). Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., and Ives, Z. (2007). DBpedia: a nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Backstrom, L. and Leskovec, J. (2011). Supervised random walks: Predicting and recommending links in social networks. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 635–644, New York, NY, USA. ACM.

Bada, M., Eckert, M., Evans, D., Garcia, K., Shipley, K., Sitnikov, D., Baumgartner, W. A., Cohen, K. B., Verspoor, K., Blake, J. A., et al. (2012). Concept annotation in the CRAFT corpus. *BMC Bioinformatics*, 13(1):1.

Baker, S., Ali, I., Silins, I., Pyysalo, S., Guo, Y., Högberg, J., Stenius, U., and Korhonen, A. (2017a). Cancer hallmarks analytics tool (CHAT): a text mining approach to organize and evaluate scientific literature on cancer. *Bioinformatics*, 33(24):3973–3981.

Baker, S., Korhonen, A.-L., and Pyysalo, S. (2017b). Cancer hallmark text classification using convolutional neural networks.

Baker, S., Silins, I., Guo, Y., Ali, I., Högberg, J., Stenius, U., and Korhonen, A. (2015). Automatic semantic classification of scientific literature according to the hallmarks of cancer. *Bioinformatics*, 32(3):432–440.

Bakker, B. and Heskes, T. (2003). Task Clustering and Gating for Bayesian Multitask Learning. *Journal of Machine Learning Research*, 4:83–99.

Banerjee, R., Choi, Y., Piyush, G., Naik, A., and Ramakrishnan, I. (2014). Automated suggestion of tests for identifying likelihood of adverse drug events. In *2014 IEEE International Conference on Healthcare Informatics (ICHI)*, pages 170–175. IEEE.

Bard, J., Rhee, S. Y., and Ashburner, M. (2005). An ontology for cell types. *Genome Biology*, 6(2):1.

Barnickel, T., Weston, J., Collobert, R., Mewes, H.-W., and Stümpflen, V. (2009). Large scale application of neural network based semantic role labeling for automated relation extraction from biomedical texts. *PLoS One*, 4(7):e6393.

Batista-Navarro, R., Rak, R., and Ananiadou, S. (2015). Optimising chemical named entity recognition with pre-processing analytics, knowledge-rich features and heuristics. *Journal of Cheminformatics*, 7(1):1.

Benchettara, N., Kanawati, R., and Rouveirol, C. (2010). Supervised machine learning applied to link prediction in bipartite social networks. In *2010 International Conference on Advances in Social Networks Analysis and Mining*, pages 326–330.

Bengio, Y., Ducharme, R., Vincent, P., and Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, pages 1137–1155.

Bengio, Y., Simard, P., and Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166.

Bingel, J. and Søgaard, A. (2017). Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, volume 2, pages 164–169.

Bodenreider, O. (2004). The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32(suppl_1):D267–D270.

Cameron, D., Bodenreider, O., Yalamanchili, H., Danh, T., Vallabhaneni, S., Thirunarayan, K., Sheth, A. P., and Rindflesch, T. C. (2013). A graph-based recovery and decomposition of Swanson's hypothesis using semantic predications. *Journal of Biomedical Informatics*, 46(2):238–251.

Cameron, D., Kavuluru, R., Rindflesch, T. C., Sheth, A. P., Thirunarayan, K., and Bodenreider, O. (2015). Context-driven automatic subgraph creation for literature-based discovery. *Journal of Biomedical Informatics*, 54:141–157.

Campos, D., Matos, S., and Oliveira, J. L. (2013). Gimli: open source and high-performance biomedical name recognition. *BMC Bioinformatics*, 14(1):54.

Campos, D., Matos, S., and Oliveira, J. L. (2015). A document processing pipeline for annotating chemical entities in scientific documents. *Journal of Cheminformatics*, 7(1):S7.

Caruana, R. (1993). Multitask learning: A knowledge-based source of inductive bias. In *Machine Learning: Proceedings of the Tenth International Conference*, pages 41–48.

Caruana, R. (1997). Multitask learning. *Mach. Learn.*, 28(1):41–75.

Chatr-aryamontri, A., Oughtred, R., Boucher, L., Rust, J., Chang, C., Kolas, N. K., O'Donnell, L., Oster, S., Theesfeld, C., Sellam, A., et al. (2017). The BioGRID interaction database: 2017 update. *Nucleic Acids Research*, 45(D1):D369–D379.

Chen, W., Liu, T.-Y., Lan, Y., Ma, Z.-M., and Li, H. (2009). Ranking measures and loss functions in learning to rank. In *Advances in Neural Information Processing Systems*, pages 315–323.

Chiu, B., Crichton, G., Korhonen, A., and Pyysalo, S. (2016a). How to train good word embeddings for biomedical NLP. *ACL 2016*, page 166.

Chiu, B., Korhonen, A., and Pyysalo, S. (2016b). Intrinsic evaluation of word vectors fails to predict extrinsic performance. *Proceedings of RepEval 2016*.

Chiu, J. P. and Nichols, E. (2016). Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Cho, K., van Merriënboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using RNN Encoder–Decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Chollet, F. et al. (2015). Keras. https://keras.io.

Cohen, T., Schvaneveldt, R., and Widdows, D. (2010). Reflective random indexing and indirect inference: A scalable method for discovery of implicit connections. *Journal of Biomedical Informatics*, 43(2):240–256.

Cohen, T., Widdows, D., and Rindflesch, T. (2014). Expansion-by-analogy: A vector symbolic approach to semantic search. In *International Symposium on Quantum Interaction*, pages 54–66. Springer.

Cohen, T., Widdows, D., Schvaneveldt, R. W., Davies, P., and Rindflesch, T. C. (2012). Discovering discovery patterns with predication-based semantic indexing. *Journal of Biomedical Informatics*, 45(6):1049–1065.

Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of ICML*, pages 160–167. ACM.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural language processing (almost) from scratch. *The Journal of Machine Learning Research*, 12:2493–2537.

Comeau, D. C., Doğan, R. I., Ciccarese, P., Cohen, K. B., Krallinger, M., Leitner, F., Lu, Z., Peng, Y., Rinaldi, F., Torii, M., et al. (2013). Bioc: a minimalist approach to interoperability for biomedical text processing. *Database*, 2013:bat064.

Consortium, G. O. et al. (2004). The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Research*, 32(suppl 1):D258–D261.

Crichton, G. (2013). Ovaz: A concept-based exploratory search system with peer-to-peer capabilities. Master's thesis, University of the West Indies, Cave Hill, Barbados.

Crichton, G., Guo, Y., Pyysalo, S., and Korhonen, A. (2018). Neural networks for link prediction in realistic biomedical graphs: a multi-dimensional evaluation of graph embedding-based approaches. *BMC Bioinformatics*, 19(1):176.

Crichton, G., Pyysalo, S., Chiu, B., and Korhonen, A. (2017). A neural network multi-task learning approach to biomedical Named Entity Recognition. *BMC Bioinformatics*, 18(1):368.

de Matos, P., Dekker, A., Ennis, M., Hastings, J., Haug, K., Turner, S., and Steinbeck, C. (2010). ChEBI: a chemistry ontology and database. *Journal of Cheminformatics*, 2:1–1.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Degtyarenko, K., De Matos, P., Ennis, M., Hastings, J., Zbinden, M., McNaught, A., Alcántara, R., Darsow, M., Guedj, M., and Ashburner, M. (2008). ChEBI: a database and ontology for chemical entities of biological interest. *Nucleic Acids Research*, 36(suppl 1):D344–D350.

DeNicola, G. M., Karreth, F. A., Humpton, T. J., Gopinathan, A., Wei, C., Frese, K., Mangal, D., Kenneth, H. Y., Yeo, C. J., Calhoun, E. S., et al. (2011). Oncogene-induced Nrf2 transcription promotes ROS detoxification and tumorigenesis. *Nature*, 475(7354):106.

DiGiacomo, R. A., Kremer, J. M., and Shah, D. M. (1989). Fish-oil dietary supplementation in patients with Raynaud's phenomenon: a double-blind, controlled, prospective study. *The American Journal of Medicine*, 86(2):158–164.

Ding, Y., Song, M., Han, J., Yu, Q., Yan, E., Lin, L., and Chambers, T. (2013). Entitymetrics: Measuring the impact of entities. *PloS one*, 8(8):e71416.

Doğan, R. I., Leaman, R., and Lu, Z. (2014). NCBI disease corpus: a resource for disease name recognition and concept normalization. *Journal of Biomedical Informatics*, 47:1–10.

Dong, X., Gabrilovich, E., Heitz, G., Horn, W., Lao, N., Murphy, K., Strohmann, T., Sun, S., and Zhang, W. (2014). Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–610. ACM.

Eronen, L., Hintsanen, P., and Toivonen, H. (2012). Biomine: a network-structured resource of biological entities for link prediction. In *Bisociative Knowledge Discovery*, pages 364–378. Springer.

Evgeniou, T., Micchelli, C. A., and Pontil, M. (2005). Learning multiple tasks with kernel methods. *J. Mach. Learn. Res.*, 6:615–637.

Federhen, S. (2012). The NCBI taxonomy database. *Nucleic Acids Research*, 40(D1):D136–D143.

Fernández, J., Gutiérrez, Y., Gómez, J. M., and Martınez-Barco, P. (2014). GPLSI: Supervised sentiment analysis in Twitter using skipgrams. In *Proceedings of SemEval*, pages 294–299.

Gaffen, S. L. and McGeachy, M. J. (2015). Integrating p38α MAPK immune signals in nonimmune cells. *Sci. Signal.*, 8(366):fs5–fs5.

Gaudet, P., Michel, P.-A., Zahn-Zabal, M., Cusin, I., Duek, P. D., Evalet, O., Gateau, A., Gleizes, A., Pereira, M., Teixeira, D., et al. (2015). The neXtProt knowledgebase on human proteins: current status. *Nucleic Acids Research*, 43(D1):D764–D770.

Gerner, M., Nenadic, G., and Bergman, C. M. (2010). LINNAEUS: a species name identification system for biomedical literature. *BMC Bioinformatics*, 11(1):1.

Goodwin, J. C., Cohen, T., and Rindflesch, T. (2012). Discovery by scent: Discovery browsing system based on the information foraging theory. In *Bioinformatics and Biomedicine Workshops (BIBMW), 2012 IEEE International Conference on*, pages 232–239. IEEE.

Gordon, M. D. and Awad, N. F. (2008). The tip of the iceberg: the quest for innovation at the base of the pyramid. In *Literature-based Discovery*, pages 23–37. Springer.

Gordon, M. D. and Dumais, S. (1998). Using latent semantic indexing for literature based discovery. *Journal of the American Society for Information Science*, 49(8):674–685.

Gordon, M. D. and Lindsay, R. K. (1996). Toward discovery support systems: A replication, re-examination, and extension of Swanson's work on literature-based discovery of a connection between Raynaud's and fish oil. *Journal of the American Society for Information Science*, 47(2):116–128.

Goyal, P. and Ferrara, E. (2018a). GEM: A Python package for graph embedding methods. *Journal of Open Source Software*, 3(29):876.

Goyal, P. and Ferrara, E. (2018b). Graph embedding techniques, applications, and performance: A survey. *Knowledge-Based Systems*, 151:78–94.

Grover, A. and Leskovec, J. (2016). node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Gulec, F. M., Bicakci, T., Sezer, E. A., Sever, H., and Raghavan, V. V. (2010). Analyzing the effectiveness of pruning and grouping methods used in literature-based discovery tools. In *Web Intelligence and Intelligent Agent Technology (WI-IAT), 2010 IEEE/WIC/ACM International Conference on*, volume 3, pages 304–308. IEEE.

Günther, S., Kuhn, M., Dunkel, M., Campillos, M., Senger, C., Petsalaki, E., Ahmed, J., Urdiales, E. G., Gewiess, A., Jensen, L. J., et al. (2008). SuperTarget and Matador: resources for exploring drug-target relationships. *Nucleic Acids Research*, 36(suppl 1):D919–D922.

Hagberg, A., Swart, P., and S Chult, D. (2008). Exploring network structure, dynamics, and function using NetworkX. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States).

Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034.

Hamosh, A., Scott, A. F., Amberger, J. S., Bocchini, C. A., and McKusick, V. A. (2005). Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Research*, 33(suppl_1):D514–D517.

Hanahan, D. and Weinberg, R. A. (2000). The hallmarks of cancer. *Cell*, 100(1):57–70.

Henry, S. and McInnes, B. T. (2017). Literature based discovery: models, methods, and trends. *Journal of Biomedical Informatics*, 74:20–32.

Hill, F., Reichart, R., and Korhonen, A. (2015). SimLex-999: Evaluating semantic models with (genuine) similarity estimation. *Computational Linguistics*.

Hoare, M., Ito, Y., Kang, T.-W., Weekes, M. P., Matheson, N. J., Patten, D. A., Shetty, S., Parry, A. J., Menon, S., Salama, R., et al. (2016). NOTCH1 mediates a switch between two distinct secretomes during senescence. *Nature Cell Biology*, 18(9):979.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hristovski, D., Džeroski, S., Peterlin, B., and Rožić, A. (2000). Supporting discovery in medicine by association rule mining of bibliographic databases. In *European Conference on Principles of Data Mining and Knowledge Discovery*, pages 446–451. Springer.

Hristovski, D., Friedman, C., Rindflesch, T. C., and Peterlin, B. (2006). Exploiting semantic relations for literature-based discovery. In *AMIA annual symposium proceedings*, volume 2006, page 349. American Medical Informatics Association.

Hristovski, D., Kastrin, A., Peterlin, B., and Rindflesch, T. C. (2010). Combining semantic relations and DNA microarray data for novel hypotheses generation. In *Linking literature, information, and knowledge for biology*, pages 53–61. Springer.

Hristovski, D., Kastrin, A., and Rindflesch, T. C. (2015). Semantics-based cross-domain collaboration recommendation in the life sciences: Preliminary results. In *Advances in Social Networks Analysis and Mining (ASONAM), 2015 IEEE/ACM International Conference on*, pages 805–806. IEEE.

Hristovski, D., Peterlin, B., Mitchell, J. A., Humphrey, S. M., Sitbon, L., and Turner, I. (2003). Improving literature based discovery support by genetic knowledge integration. *Stud Health Technol Inform*, 95.

Hristovski, D., Rindflesch, T., and Peterlin, B. (2013). Using literature-based discovery to identify novel therapeutic approaches. *Cardiovascular & Hematological Agents in Medicinal Chemistry (Formerly Current Medicinal Chemistry-Cardiovascular & Hematological Agents)*, 11(1):14–24.

Hristovski, D., Stare, J., Peterlin, B., and Dzeroski, S. (2001). Supporting discovery in medicine by association rule mining in Medline and UMLS. *Studies in Health Technology and Informatics*, (2):1344–1348.

Hu, X., Zhang, X., Yoo, I., and Zhang, Y. (2006). A semantic approach for mining hidden links from complementary and non-interactive biomedical literature. In *Proceedings of the 2006 SIAM International Conference on Data Mining*, pages 200–209. SIAM.

Huang, P.-S., He, X., Gao, J., Deng, L., Acero, A., and Heck, L. (2013). Learning deep structured semantic models for web search using clickthrough data. In *CIKM*, CIKM '13, pages 2333–2338, New York, NY, USA. ACM.

Huang, Z., Li, X., and Chen, H. (2005). Link prediction approach to collaborative filtering. In *Proceedings of the 5th ACM/IEEE-CS Joint Conference on Digital Libraries*, JCDL '05, pages 141–142, New York, NY, USA. ACM.

Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Hunter, L. and Cohen, K. B. (2006). Biomedical language processing: What's beyond PubMed? *Molecular Cell*, 21(5):589 – 594.

Irsoy, O. and Cardie, C. (2014). Opinion mining with deep recurrent neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 720–728.

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bull Soc Vaudoise Sci Nat*, 37:547–579.

Kastrin, A., Rindflesch, T. C., and Hristovski, D. (2014). Link prediction on the semantic MEDLINE network. In *International Conference on Discovery Science*, pages 135–143. Springer.

Kastrin, A., Rindflesch, T. C., Hristovski, D., et al. (2016). Link prediction on a network of co-occurring MeSH terms: towards literature-based discovery. *Methods of Information in Medicine*, 55(4):340–346.

Katukuri, J. R., Xie, Y., Raghavan, V. V., and Gupta, A. (2012). Hypotheses generation as supervised link discovery with automated class labeling on large-scale biomedical concept networks. *BMC Genomics*, 13(3):S5.

Kilicoglu, H., Shin, D., Fiszman, M., Rosemblat, G., and Rindflesch, T. C. (2012). SemMedDB: a PubMed-scale repository of biomedical semantic predications. *Bioinformatics*, 28(23):3158–3160.

Kim, J.-D., Ohta, T., Pyysalo, S., Kano, Y., and Tsujii, J. (2009). Overview of Bionlp'09 shared task on event extraction. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing: Shared Task*, pages 1–9. Association for Computational Linguistics.

Kim, J.-D., Ohta, T., and Tsujii, J. (2008). Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics*, 9(1):1.

Kim, J.-D., Ohta, T., Tsuruoka, Y., Tateisi, Y., and Collier, N. (2004). Introduction to the bio-entity recognition task at JNLPBA. In *Proceedings of JNLPBA*, pages 70–75.

Kim, J.-D., Pyysalo, S., Ohta, T., Bossy, R., Nguyen, N., and Tsujii, J. (2011). Overview of Bionlp shared task 2011. In *Proceedings of the BioNLP Shared Task 2011 Workshop*, pages 1–6.

Kim, J.-D., Wang, Y., and Yasunori, Y. (2013). The genia event extraction shared task, 2013 edition-overview. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 8–15. Association for Computational Linguistics.

Kim, Y. H., Choi, Y. W., Lee, J., Soh, E. Y., Kim, J.-H., and Park, T. J. (2017). Senescent tumor cells lead the collective invasion in thyroid cancer. *Nature Communications*, 8:15208.

Kingma, D. P. and Ba, J. (2015). Adam: A method for stochastic optimization. In *Proceedings of 3rd International Conference on Learning Representations, ICLR, 2015*.

Kipf, T. N. and Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *Proceedings of 5th International Conference on Learning Representations, ICLR, 2017*.

Korhonen, A., Guo, Y., Baker, S., Yetisgen-Yildiz, M., Stenius, U., Narita, M., and Liò, P. (2014). Improving literature-based discovery with advanced text mining. In *International Meeting on Computational Intelligence Methods for Bioinformatics and Biostatistics*, pages 89–98. Springer.

Kosmopoulos, A., Androutsopoulos, I., and Paliouras, G. (2015). Biomedical semantic indexing using dense word vectors in BioASQ. *Journal Of Biomedical Semantics*.

Kostoff, R. N. (2008a). Literature-related discovery (LRD): introduction and background. *Technological Forecasting and Social Change*, 75(2):165–185.

Kostoff, R. N. (2008b). Literature-related discovery (LRD): Potential treatments for cataracts. *Technological Forecasting and Social Change*, 75(2):215–225.

Kostoff, R. N. (2010). Literature-related discovery: common factors for Parkinson's disease and Crohn's disease. Technical report, MITRE CORP MCLEAN VA.

Kostoff, R. N. and Briggs, M. B. (2008). Literature-related discovery (LRD): potential treatments for Parkinson's disease. *Technological Forecasting and Social Change*, 75(2):226–238.

Kostoff, R. N., Briggs, M. B., and Lyons, T. J. (2008a). Literature-related discovery (LRD): Potential treatments for multiple sclerosis. *Technological Forecasting and Social Change*, 75(2):239–255.

Kostoff, R. N., Briggs, M. B., Solka, J. L., and Rushenberg, R. L. (2008b). Literature-related discovery (LRD): Methodology. *Technological Forecasting and Social Change*, 75(2):186–202.

Kostoff, R. N., Solka, J. L., Rushenberg, R. L., and Wyatt, J. A. (2008c). Literature-related discovery (LRD): water purification. *Technological Forecasting and Social Change*, 75(2):256–275.

Krallinger, M., Leitner, F., Rabal, O., Vazquez, M., Oyarzabal, J., and Valencia, A. (2015). CHEMDNER: The drugs and chemical names extraction challenge. *J. Cheminformatics*, 7(S-1):S1.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition.

Lapesa, G. and Evert, S. (2014). A large scale evaluation of distributional semantic models: Parameters, interactions and model selection. *Transactions of the Association for Computational Linguistics*, 2:531–545.

Leaman, R. and Gonzalez, G. (2008). BANNER: an executable survey of advances in biomedical named entity recognition. In *Proceedings of PSB*, volume 13, pages 652–663.

Leaman, R., Miller, C., and Gonzalez, G. (2009). Enabling recognition of diseases in biomedical text with machine learning: corpus and benchmark. In *Proceedings of the 2009 Symposium on Languages in Biology and Medicine*, volume 82.

LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436.

Leskovec, J., Huttenlocher, D., and Kleinberg, J. (2010). Predicting positive and negative links in online social networks. In *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, pages 641–650, New York, NY, USA. ACM.

Levy, O., Goldberg, Y., and Dagan, I. (2015). Improving distributional similarity with lessons learned from word embeddings. *TACL*, 3:211–225.

Liben-Nowell, D. and Kleinberg, J. (2003). The link prediction problem for social networks. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, CIKM '03, pages 556–559, New York, NY, USA. ACM.

Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. In *AAAI*, volume 15, pages 2181–2187.

Lin, Y., Shen, S., Liu, Z., Luan, H., and Sun, M. (2016). Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (volume 1: Long Papers)*, volume 1, pages 2124–2133.

Lindsay, R. K. and Gordon, M. D. (1999). Literature-based discovery by lexical statistics. *Journal of the American Society for Information Science*, 50(7):574–587.

Lipscomb, C. E. (2000). Medical subject headings (MeSH). *Bulletin of the Medical Library Association*, 88(3):265.

Liu, X., Gao, J., He, X., Deng, L., Duh, K., and Wang, Y.-Y. (2015). Representation Learning Using Multi-Task Deep Neural Networks for Semantic Classification and Information Retrieval. *NAACL*.

Lu, Y., Guo, Y., and Korhonen, A. (2017). Link prediction in drug-target interactions network using similarity indices. *BMC Bioinformatics*, 18(1):39.

Luong, M., Le, Q. V., Sutskever, I., Vinyals, O., and Kaiser, L. (2016). Multi-task sequence to sequence learning. In *Proceedings of 4th International Conference on Learning Representations, ICLR, 2016*.

Lussier, Y. and Friedman, C. (2007). BiomedLEE: a natural-language processor for extracting and representing phenotypes, underlying molecular mechanisms and their relationships. *ISMB: 2007*.

Maaten, L. v. d. and Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9(Nov):2579–2605.

Maglott, D., Ostell, J., Pruitt, K. D., and Tatusova, T. (2005). Entrez Gene: gene-centered information at NCBI. *Nucleic Acids Research*, 33(suppl 1):D54–D58.

Marchionini, G. (2006). Exploratory search: from finding to understanding. *Communications of the ACM*, 49(4):41–46.

Maurer, A., Pontil, M., and Romera-Paredes, B. (2016). The benefit of multitask representation learning. *J. Mach. Learn. Res.*, 17(1):2853–2884.

McDonald, D., McNicoll, I., Weir, G., Reimer, T., Redfearn, J., Jacobs, N., and Bruce, R. (2012). The value and benefits of text mining. *JISC Digital Infrastructure*.

McKinney, W. et al. (2010). Data structures for statistical computing in Python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX.

Mi, H. and Thomas, P. (2009). Panther pathway: an ontology-based pathway database coupled with data analysis tools. *Protein Networks and Pathway Analysis*, pages 123–140.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, pages 3111–3119.

Miller, G. A. (1995). WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41.

Mower, J., Subramanian, D., Shang, N., and Cohen, T. (2016). Classification-by-analogy: using vector representations of implicit relationships to identify plausibly causal drug/side-effect relationships. In *AMIA Annual Symposium Proceedings*, volume 2016, page 1940. American Medical Informatics Association.

Muneeb, T., Sahu, S. K., and Anand, A. (2015). Evaluating distributed word representations for capturing semantics of biomedical concepts. In *Proceedings of ACL-IJCNLP*, page 158.

Mungall, C. J., Torniai, C., Gkoutos, G. V., Lewis, S. E., and Haendel, M. A. (2012). Uberon, an integrative multi-species anatomy ontology. *Genome Biology*, 13(1):1.

Munkhdalai, T., Li, M., Batsuren, K., Park, H. A., Choi, N. H., and Ryu, K. H. (2015). Incorporating domain knowledge in chemical and biomedical named entity recognition with word representations. *Journal of Cheminformatics*, 7(1):1.

Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted Boltzmann machines. In *Proceedings of ICML-10*, pages 807–814.

Newman, M. E. (2001). Clustering and preferential attachment in growing networks. *Physical Review E*, 64(2):025102.

Nguyen, T. H. and Grishman, R. (2015). Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48.

Nickel, M., Murphy, K., Tresp, V., and Gabrilovich, E. (2016). A review of relational machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1):11–33.

Ogren, P. V. (2006). Knowtator: a protégé plug-in for annotated corpus construction. In *Proceedings of NAACL-HTL 2006*, pages 273–275.

Ohta, T., Pyysalo, S., Rak, R., Rowley, A., Chun, H.-W., Jung, S.-J., Jeong, C.-h., Choi, S.-p., and Ananiadou, S. (2013). Overview of the pathway curation (PC) task of bioNLP shared task 2013. In *Proceedings of the BioNLP Shared Task 2013 Workshop*, pages 67–75. Association for Computational Linguistics.

Ohta, T., Pyysalo, S., Tsujii, J., and Ananiadou, S. (2012). Open-domain anatomical entity mention detection. In *Proceedings of the workshop on detecting structure in scholarly discourse*, pages 27–36. Association for Computational Linguistics.

Ohta, T., Tateisi, Y., and Kim, J.-D. (2002). The GENIA corpus: An annotated research abstract corpus in molecular biology domain. In *Proceedings of HTL*, pages 82–86.

Oliphant, T. E. (2006). *A guide to NumPy*, volume 1. Trelgol Publishing USA.

Ou, M., Cui, P., Pei, J., Zhang, Z., and Zhu, W. (2016). Asymmetric transitivity preserving graph embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1105–1114, New York, NY, USA. ACM.

Pakhomov, S., McInnes, B., Adam, T., Liu, Y., Pedersen, T., and Melton, G. B. (2010). Semantic similarity and relatedness between clinical terms: an experimental study. In *Proceedings of AMIA*, volume 2010, page 572.

Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., and Lerer, A. (2017). Automatic differentiation in pytorch. In *NIPS-W*.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global Vectors for Word Representation. In *Proceedings of EMNLP*, volume 14, pages 1532–1543.

Perozzi, B., Al-Rfou, R., and Skiena, S. (2014). DeepWalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 701–710, New York, NY, USA. ACM.

Preiss, J. and Stevenson, M. (2016). The effect of word sense disambiguation accuracy on literature based discovery. *BMC Medical Informatics and Decision Making*, 16(1):57.

Preiss, J. and Stevenson, M. (2017). Quantifying and filtering knowledge generated by literature based discovery. *BMC Bioinformatics*, 18(7):249.

Preiss, J., Stevenson, M., et al. (2018). HiDE: a tool for unrestricted literature based discovery. *Demo Proceedings, COLING 2018*, pages 34–36.

Preiss, J., Stevenson, M., and Gaizauskas, R. (2015). Exploring relation types for literature-based discovery. *Journal of the American Medical Informatics Association*, 22:987–992.

Preiss, J., Stevenson, M., and McClure, M. H. (2012). Towards semantic literature based discovery. In *2012 AAAI Fall Symposium Series: Information Retrieval and Knowledge Discovery in Biomedical Text*, volume 30, pages 7–18.

Pruitt, K. D. and Maglott, D. R. (2001). RefSeq and LocusLink: NCBI gene-centered resources. *Nucleic Acids Research*, 29(1):137–140.

Pyysalo, S. and Ananiadou, S. (2013). Anatomical entity mention recognition at literature scale. *Bioinformatics*, 30(6):868–875.

Pyysalo, S., Baker, S., Ali, I., Haselwimmer, S., Shah, T., Young, A., Guo, Y., Högberg, J., Stenius, U., Narita, M., and Korhonen, A. (2018). LION LBD: a literature-based discovery system for cancer biology. *Bioinformatics*, page bty845.

Pyysalo, S., Ginter, F., Moen, H., Salakoski, T., and Ananiadou, S. (2013). Distributional semantics resources for biomedical text processing. *Proceedings of LBM*.

Pyysalo, S., Ohta, T., Miwa, M., Cho, H.-C., Tsujii, J., and Ananiadou, S. (2012a). Event extraction across multiple levels of biological organization. *Bioinformatics*, 28(18):i575–i581.

Pyysalo, S., Ohta, T., Miwa, M., and Tsujii, J. (2011). Towards exhaustive protein modification event extraction. In *Proceedings of BioNLP 2011 Workshop*, pages 114–123. Association for Computational Linguistics.

Pyysalo, S., Ohta, T., Rak, R., Rowley, A., Chun, H.-W., Jung, S.-J., Choi, S.-P., Tsujii, J., and Ananiadou, S. (2015). Overview of the Cancer Genetics and Pathway Curation tasks of BioNLP shared task 2013. *BMC Bioinformatics*, 16(10):1.

Pyysalo, S., Ohta, T., Rak, R., Sullivan, D., Mao, C., Wang, C., Sobral, B., Tsujii, J., and Ananiadou, S. (2012b). Overview of the ID, EPI and REL tasks of BioNLP shared task 2011. *BMC Bioinformatics*, 13(11):1.

Rastegar-Mojarad, M., Elayavilli, R. K., Wang, L., Prasad, R., and Liu, H. (2016). Prioritizing adverse drug reaction and drug repositioning candidates generated by literature-based discovery. In *Proceedings of the 7th ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 289–296. ACM.

Rei, M., Crichton, G., and Pyysalo, S. (2016). Attending to characters in neural sequence labeling models. In *Proceedings of COLING 2016*, pages 309–318.

Rei, M. and Yannakoudakis, H. (2016). Compositional sequence labeling models for error detection in learner writing.

Rindflesch, T. C. and Fiszman, M. (2003). The interaction of domain knowledge and linguistic structure in natural language processing: interpreting hypernymic propositions in biomedical text. *Journal of Biomedical Informatics*, 36(6):462–477.

Rosse, C. and Mejino Jr, J. L. (2008). The foundational model of anatomy ontology. In *Anatomy Ontologies for Bioinformatics*, pages 59–117. Springer.

Rosse, C., Mejino Jr, J. L., et al. (2003). A reference ontology for biomedical informatics: the foundational model of anatomy. *Journal of Biomedical Informatics*, 36(6):478–500.

Ruder, S. (2017). An overview of multi-task learning in deep neural networks. *arXiv preprint arXiv:1706.05098*.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, 323(6088):533.

Sasaki, Y., Tsuruoka, Y., McNaught, J., and Ananiadou, S. (2008). How to make the most of NE dictionaries in statistical NER. *BMC Bioinformatics*, 9(11):1.

Schlichtkrull, M., Kipf, T. N., Bloem, P., van den Berg, R., Titov, I., and Welling, M. (2018). Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*, pages 593–607. Springer.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117.

Schuler, K. K. (2005). VerbNet: A broad-coverage, comprehensive verb lexicon.

Sebastian, Y., Siew, E.-G., and Orimaye, S. O. (2015). *Predicting Future Links Between Disjoint Research Areas Using Heterogeneous Bibliographic Information Network*, pages 610–621. Springer International Publishing, Cham.

Sebastian, Y., Siew, E.-G., and Orimaye, S. O. (2017a). Emerging approaches in literature-based discovery: techniques and performance review. *The Knowledge Engineering Review*, 32.

Sebastian, Y., Siew, E.-G., and Orimaye, S. O. (2017b). Learning the heterogeneous bibliographic information network for literature-based discovery. *Knowledge-Based Systems*, 115:66–79.

Settles, B. (2005). ABNER: An open source tool for automatically tagging genes, proteins, and other entity names in text. *Bioinformatics*, 21(14):3191–3192.

Shang, N., Xu, H., Rindflesch, T. C., and Cohen, T. (2014). Identifying plausible adverse drug reactions using knowledge extracted from the literature. *Journal of Biomedical Informatics*, 52:293–310.

Simonyan, K. and Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *Proceedings of 3rd International Conference on Learning Representations, ICLR, 2015*.

Simpson, M. S. and Demner-Fushman, D. (2012). *Biomedical Text Mining: A Survey of Recent Progress*, pages 465–517. Springer US, Boston, MA.

Smalheiser, N. R. (2012). Literature-based discovery: Beyond the ABCs. *Journal of the American Society for Information Science and Technology*, 63(2):218–224.

Smalheiser, N. R. and Swanson, D. R. (1996a). Indomethacin and Alzheimer's disease. *Neurology*, 46(2):583–583.

Smalheiser, N. R. and Swanson, D. R. (1996b). Linking estrogen to Alzheimer's disease an informatics approach. *Neurology*, 47(3):809–810.

Smalheiser, N. R. and Swanson, D. R. (1998). Calcium-independent Phospholipase A2 and Schizophrenia. *Archives of General Psychiatry*, 55(8):752–753.

Smith, L., Tanabe, L. K., nee Ando, R. J., Kuo, C.-J., Chung, I.-F., Hsu, C.-N., Lin, Y.-S., Klinger, R., Friedrich, C. M., Ganchev, K., et al. (2008). Overview of BioCreative II gene mention recognition. *Genome Biology*, 9(Suppl 2):1–19.

Søgaard, A. and Goldberg, Y. (2016). Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the ACL*, page 231.

Srinivasan, P. (2004). Text mining: generating hypotheses from medline. *Journal of the American Society for Information Science and Technology*, 55(5):396–413.

Srinivasan, P. and Libbus, B. (2004). Mining MEDLINE for implicit links between dietary substances and diseases. *Bioinformatics*, 20(suppl_1):i290–i296.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.*, 15(1):1929–1958.

Stark, C., Breitkreutz, B.-J., Reguly, T., Boucher, L., Breitkreutz, A., and Tyers, M. (2006). BioGRID: a general repository for interaction datasets. *Nucleic Acids Research*, 34(suppl 1):D535–D539.

Stegmann, J. and Grohmann, G. (2003). Hypothesis generation guided by co-word clustering. *Scientometrics*, 56(1):111–135.

Stenetorp, P., Soyer, H., Pyysalo, S., Ananiadou, S., and Chikayama, T. (2012). Size (and domain) matters: Evaluating semantic word space representations for biomedical text. In *Proceedings of SMBM*.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems*, pages 3104–3112.

Swanson, D. R. (1986a). Fish oil, Raynaud's syndrome, and undiscovered public knowledge. *Perspectives in Biology and Medicine*, 30(1):7–18.

Swanson, D. R. (1986b). Undiscovered public knowledge. *The Library Quarterly*, 56(2):103–118.

Swanson, D. R. (1988). Migraine and Magnesium: eleven neglected connections. *Perspectives in Biology and Medicine*, 31(4):526–557.

Swanson, D. R. (1990a). Medical literature as a potential source of new knowledge. *Bulletin of the Medical Library Association*, 78(1):29.

Swanson, D. R. (1990b). Somatomedin C and Arginine: implicit connections between mutually isolated literatures. *Perspectives in Biology and Medicine*, 33(2):157–186.

Swanson, D. R. (2008). Literature-based discovery? The very idea. In *Literature-based discovery*, pages 3–11. Springer.

Swanson, D. R. and Smalheiser, N. R. (1997). An interactive system for finding complementary literatures: a stimulus to scientific discovery. *Artificial Intelligence*, 91(2):183–203.

Swanson, D. R., Smalheiser, N. R., and Torvik, V. I. (2006). Ranking indirect connections in literature-based discovery: The role of medical subject headings. *Journal of the American Society for Information Science and Technology*, 57(11):1427–1439.

Symonds, M., Bruza, P., and Sitbon, L. (2014). The efficiency of corpus-based distributional models for literature-based discovery on large data sets. In *Proceedings of the Second Australasian Web Conference-volume 155*, pages 49–57. Australian Computer Society, Inc.

Tanabe, L., Xie, N., Thom, L. H., Matten, W., and Wilbur, W. J. (2005). GENETAG: a tagged corpus for gene/protein named entity recognition. *BMC Bioinformatics*, 6(1):1.

Tang, J., Qu, M., Wang, M., Zhang, M., Yan, J., and Mei, Q. (2015). LINE: Large-scale Information Network Embedding. In *Proceedings of WWW 2015*. ACM.

Thaicharoen, S., Altman, T., Gardiner, K., and Cios, K. J. (2009). Discovering relational knowledge from two disjoint sets of literatures using inductive logic programming. In *2009 IEEE Symposium on Computational Intelligence and Data Mining*, pages 283–290. IEEE.

Torvik, V. I. and Smalheiser, N. R. (2007). A quantitative model for linking two disparate sets of articles in medline. *Bioinformatics*, 23(13):1658–1665.

Tsuruoka, Y., Miwa, M., Hamamoto, K., Tsujii, J., and Ananiadou, S. (2011). Discovering and visualizing indirect associations between biomedical concepts. *Bioinformatics*, 27(13):i111–i119.

Tsuruoka, Y., Tateishi, Y., Kim, J.-D., Ohta, T., McNaught, J., Ananiadou, S., and Tsujii, J. (2005). Developing a robust part-of-speech tagger for biomedical text. In *Proceedings of Panhellenic Conference on Informatics*, pages 382–392.

Turian, J., Ratinov, L., and Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proceedings of ACL*, pages 384–394.

Turney, P. D. (2012). Domain and function: A dual-space model of semantic relations and compositions. *Journal of Artificial Intelligence Research*, pages 533–585.

Van Der Heijden, M., Zimberlin, C. D., Nicholson, A. M., Colak, S., Kemp, R., Meijer, S. L., Medema, J. P., Greten, F. R., Jansen, M., Winton, D. J., et al. (2016). Bcl-2 is a critical mediator of intestinal transformation. *Nature Communications*, 7:10916.

Van Rossum, G. and Drake Jr, F. L. (1995). *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam.

Verspoor, K., Cohen, K. B., Lanfranchi, A., Warner, C., Johnson, H. L., Roeder, C., Choi, J. D., Funk, C., Malenkiy, Y., Eckert, M., et al. (2012). A corpus of full-text journal articles is a robust evaluation tool for revealing differences in performance of biomedical natural language processing tools. *BMC Bioinformatics*, 13(1):1.

Wang, D., Cui, P., and Zhu, W. (2016). Structural Deep Network Embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 1225–1234, New York, NY, USA. ACM.

Wang, P., Qian, Y., Soong, F. K., He, L., and Zhao, H. (2015). A unified tagging solution: Bidirectional LSTM Recurrent Neural Network with Word Embedding. *CoRR*, abs/1511.00215.

Wang, Y., Kim, J.-D., Sætre, R., Pyysalo, S., and Tsujii, J. (2009). Investigating heterogeneous protein annotations toward cross-corpora utilization. *BMC Bioinformatics*, 10(1):1.

Wang, Y. and Zeng, J. (2013). Predicting drug-target interactions using restricted Boltzmann machines. *Bioinformatics*, 29(13):i126–i134.

Weeber, M., Klein, H., Aronson, A. R., Mork, J. G., De Jong-van Den Berg, L., and Vos, R. (2000). Text-based discovery in biomedicine: the architecture of the DAD-system. In *Proceedings of the AMIA Symposium*, page 903. American Medical Informatics Association.

Weeber, M., Klein, H., de Jong-van den Berg, L. T., and Vos, R. (2001). Using concepts in literature-based discovery: Simulating Swanson's Raynaud–fish oil and migraine–magnesium discoveries. *Journal of the American Society for Information Science and Technology*, 52(7):548–557.

Weeber, M., Vos, R., Klein, H., de Jong-van den Berg, L. T., Aronson, A. R., and Molema, G. (2003). Generating hypotheses by discovering implicit associations in the literature: a case report of a search for new potential therapeutic uses for thalidomide. *Journal of the American Medical Informatics Association*, 10(3):252–259.

Wei, C.-H., Kao, H.-Y., and Lu, Z. (2012). PubTator: a PubMed-like interactive curation system for document triage and literature curation. *BioCreative 2012 Workshop*, 05.

Wei, C.-H., Kao, H.-Y., and Lu, Z. (2013). PubTator: a Web-based text mining tool for assisting Biocuration. *Nucleic Acids Research*, 41.

Wei, C.-H., Peng, Y., Leaman, R., Davis, A. P., Mattingly, C. J., Li, J., Wiegers, T. C., and Lu, Z. (2015). Overview of the Biocreative V chemical disease relation (CDR) task. In *Proceedings of the BioCreative 5 workshop*, pages 154–166.

White, R. W. and Roth, R. A. (2009). Exploratory search: Beyond the query-response paradigm. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, 1(1):1–98.

Wilkowski, B., Fiszman, M., Miller, C. M., Hristovski, D., Arabandi, S., Rosemblat, G., and Rindflesch, T. C. (2011). Graph-based methods for discovery browsing with semantic predications. In *AMIA annual symposium proceedings*, volume 2011, page 1514. American Medical Informatics Association.

Wren, J. D. (2004). Extending the mutual information measure to rank inferred literature relationships. *BMC Bioinformatics*, 5(1):145.

Wren, J. D., Bekeredjian, R., Stewart, J. A., Shohet, R. V., and Garner, H. R. (2004). Knowledge discovery by automated identification and ranking of implicit relationships. *Bioinformatics*, 20(3):389–398.

Wu, C. H., Apweiler, R., Bairoch, A., Natale, D. A., Barker, W. C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., et al. (2006). The Universal Protein Resource (UniProt): an expanding universe of protein information. *Nucleic Acids Research*, 34(suppl 1):D187–D191.

Wu, Z., Valentini-Botinhao, C., Watts, O., and King, S. (2015). Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis. In *Proceedings of ICASSP 2015*, pages 4460–4464. IEEE.

Yang, B., Yih, W.-t., He, X., Gao, J., and Deng, L. (2014). Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv:1412.6575*.

Yang, Y., Lichtenwalter, R. N., and Chawla, N. V. (2015). Evaluating link prediction methods. *Knowledge and Information Systems*, 45(3):751–782.

Yetisgen-Yildiz, M. and Pratt, W. (2008). Evaluation of literature-based discovery systems. In *Literature-based discovery*, pages 101–113. Springer.

Yetisgen-Yildiz, M. and Pratt, W. (2009). A new evaluation methodology for literature-based discovery systems. *Journal of Biomedical Informatics*, 42(4):633–643.

Zeng, D., Liu, K., Chen, Y., and Zhao, J. (2015). Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1753–1762.

Zeng, T. and Ji, S. (2015). Deep convolutional neural networks for multi-instance multi-task learning. In *Proceedings of ICDM 2015*, pages 579–588.

Zhang, R., Cairelli, M. J., Fiszman, M., Kilicoglu, H., Rindflesch, T. C., Pakhomov, S. V., and Melton, G. B. (2014). Exploiting literature-derived knowledge and semantics to identify potential prostate cancer drugs. *Cancer Informatics*, 13:CIN–S13889.

Zhou, G. and Su, J. (2004). Exploring deep knowledge resources in biomedical name recognition. In *Proceedings of JNLPBA*, pages 96–99.

Appendix A

Multi-task Learning Biomedical NER

This Appendix contains supplementary information for Chapter 3.

A.1 Details of Datasets

We used 16 biomedical corpora representing 15 NER corpora and one part-of-speech (POS) corpus. Details of their entity counts are in Table A.1. Details of their creation, prior use, and conversion into the CoNLL format used to train, develop and test our methods are presented in the following.

A.1.1 AnatEM Corpus

The extended Anatomical Entity Mention corpus Pyysalo and Ananiadou (2013) is the result of combining and extending the Anatomical Entity Mention (AnEM) corpus Ohta et al. (2012) and the Multi-level Event Extraction corpus (MLEE) Pyysalo et al. (2012a). AnEM consists of 500 randomly selected PubMed abstracts and full-text extracts annotated for anatomical entity mentions. MLEE consists of 262 PubMed abstracts on the molecular mechanisms of cancer, specifically relating to angiogenesis. MLEE is also annotated for anatomical entities specified in AnEM.

AnatEM was created by combining the anatomical entity annotations of the AnEM and MLEE corpora, then manual annotation was done on an additional 100 documents following the selection criteria of AnEM and 350 documents following those of MLEE, for a selection of topics related to cancer. The resulting corpus thus consists of 1212 documents, 600 of which are drawn randomly from abstracts and full texts as in AnEM, and the remaining 612 are a targeted selection of PubMed abstracts relating to the molecular mechanisms of cancer.

Conversion The AnatEM corpus data is available from http://nactem.ac.uk/anatomytagger/ in multiple formats, including CoNLL-style IOB, and is provided with a pre-defined split into train, development and test subsets. We use this data in a single-class NER setting, mapping all NE labels to ANATOMY, but otherwise without modification; the number of annotations and their spans are thus identical to the source data.

A.1.2 BC2GM Corpus

The BioCreative II Gene Mention (BC2GM) task corpus consists of 20,000 sentences from biomedical publication abstracts and is annotated for mentions of the names of genes, proteins and related entities using the single NE class GENE Smith et al. (2008). It has become the major NER benchmark for gene/proteins names and has been used to train and evaluate many available NER systems such as BANNER Leaman and Gonzalez (2008) and Gimli Campos et al. (2013).

Conversion The BC2GM corpus is available from http://www.biocreative.org/ in a custom standoff format and a standard train/test split. We created a development set by splitting off 2,500 sentences from the training data and converted the corpus into CoNLL format using tools available from https://github.com/spyysalo/bc2gm-corpus.

The BC2GM corpus has the unique feature of defining alternative boundaries for some of the annotated names. For the conversion, we only used the primary annotations (`GENE.eval` files), which could be represented highly accurately in the CoNLL format: the converted data contained 99.95% of the number of annotations in the original. No differences from token boundaries were introduced: all names in the converted data matched names in the source data.

A.1.3 BC4CHEMD Corpus

The BioCreative IV Chemical and Drug (BC4CHEMD) named entity recognition task corpus consists of 10,000 abstracts annotated for mentions of chemical and drug names using a single class, CHEMICAL Krallinger et al. (2015).

Conversion The BC4CHEMD corpus data is available from http://www.biocreative.org/ in a TAB-separated standoff format and defines standard training, development and test subsets. We converted the data into CoNLL format using custom tools available from https://github.com/spyysalo/chemdner-corpora, mapping non-ASCII characters to ASCII. The basic conversion is highly accurate; the number of annotations in the converted data is

99.95% of that in the source. Non-ASCII characters in the source and tokenisation differences lowered the number of matching strings somewhat, to 97.16%.

A.1.4 BC5CDR Corpus

The BioCreative V Chemical Disease Relation (CDR) corpus was created for the BioCreative V Chemical Disease Relation (CDR) Task Wei et al. (2015) and consists of human annotations of all chemicals, diseases and their interactions in 1,500 PubMed articles. 1,400 of these articles were selected from an existing 150,000 chemical-disease interactions which were annotated by CTD-Pfizer. The CTD biocurators followed CTD's rigorous curation process and curated interactions from mostly just the abstract, but referenced the full text when it was necessary to resolve relevant issues mentioned in the abstract. The remaining 100 articles were completely new.

Conversion The BC5CDR corpus is available in BioC Comeau et al. (2013) and PubTator Wei et al. (2013) formats from http://www.biocreative.org/ with pre-defined training, development and test subsets. We converted the chemical and disease annotations of the corpus from the PubTator format using tools available from https://github.com/spyysalo/pubtator. The conversion introduced only minimal divergence, increasing the annotation number by two to 100.01% of the original due to sentence splitting errors inside annotation spans. 99.94% of the annotated strings in the source match those in the converted data, reflecting rare instances where annotation boundaries occurred inside alphanumeric tokens.

A.1.5 BioNLP09 Corpus

The BioNLP'09 shared task on event extraction Kim et al. (2009) targeted semantically rich event extraction, involving the extraction of several different classes of information. To focus on these novel aspects of the event extraction task, it was assumed that NER has already been performed and the task began with a given set of gold protein annotations. The named entities in the BioNLP task data were prepared based on the GENIA event corpus. Part of the data were derived from the publicly available event corpus Kim et al. (2008), and the remainder from an unpublished portion of the corpus.

Conversion The BioNLP'09 shared task data is available from www.nactem.ac.uk/tsujii/GENIA/SharedTask/ in the `.ann` standoff format first introduced for the task. We use the PROTEIN annotations of the corpus (the only physical entity annotations released also for its test data) and the training, development and test split of the original dataset. The data

was converted from standoff to the CoNLL format using the `standoff2conll` tool available from https://github.com/spyysalo/standoff2conll.

After conversion, the number of annotations was 99.96% of the number in the source, and 99.69% of names in the original data matched names in the converted data (ignoring whitespace), indicating that almost all of the original annotations could be exactly represented in the CoNLL format with the applied tokenisation

A.1.6 BioNLP11 Corpora

Similar to the BioNLP'09 task, the BioNLP Shared Task 2011 Kim et al. (2011); Pyysalo et al. (2012b) was focused on semantically rich tasks such as Infectious Diseases (ID) and Epigenetics and Post-translational Modifications (EPI). The ID task was concerned with the molecular mechanisms of infection, virulence and resistance while the EPI task focused on the extraction of statements regarding chemical modifications of DNA and proteins. Both tasks used manual annotations created specifically for the shared task, with automatic support for the initial tagging of named entities.

The texts for the EPI task corpus were drawn from PubMed abstracts annotated with the MeSH term corresponding to the target event (e.g. Acetylation). Protein/Gene entity mentions in the selected abstracts were automatically tagged using the BANNER Leaman and Gonzalez (2008) named entity tagger trained on the GENETAG Tanabe et al. (2005) corpus. Abstracts where fewer than five entities are found were removed and documents not relevant to the targeted topic were also manually removed.

The data for the ID corpus were drawn from the primary text content of full-text PMC open access documents deemed by infectious diseases domain experts to be representative publications on two-component regulatory systems. The annotation of the Protein entities was performed automatically using NeMine Sasaki et al. (2008) trained on the JNLPBA data Kim et al. (2004) with threshold 0.05, filtered to only GENE and Protein types.

Conversion The BioNLP'11 corpora are available from http://2011.bionlp-st.org/ in the standoff format used for the BioNLP'09 data (Section A.1.5). We use the standard training, development and test sets of each of the BioNLP'11 corpora and all physical entity annotations released for all subsets of the two corpora. Conversion was performed with the `standoff2conll` tool. As the BioNLP'11 ID task data contained a large number of annotations where more than one name occurred inside the span of another annotation (e.g. REGULON-OPERON or TWO-COMPONENT-SYSTEM), we resolved overlaps in favour of

keeping the shorter of any pair of overlapping annotations,[1] thus maximizing the number of annotations carried over from the source. Notably, this overlap pattern occurred for all 492 TWO-COMPONENT-SYSTEM annotations in the corpus (3.8% of all annotations), leading to the elimination of this annotation type from the converted data.

The converted EPI data contains 99.87% of the number of annotations in the source, but just 94.86% of originals matched converted in text, reflecting a comparatively high number of cases where an annotation boundary occurred within an alphanumeric token. For ID, the number of annotations fell to 86.99% in conversion, reflecting the frequent pattern of annotation overlap. The fraction of matching names was 85.53%, indicating that annotation boundaries rarely differ from token boundaries.

A.1.7 BioNLP13 Corpora

The BioNLP 2013 Shared Task focused on knowledge-based construction. There were six tasks in this Shared Task, of which three datasets were used for our work: GENIA Event Extraction (GE), Cancer Genetics (CG) and Pathway Curation (PC).

The GE corpus consists of 20 full paper articles sourced from PubMed Central Open Access subset (PMCOA) with 7721 spans manually annotated as protein names Kim et al. (2013). The CG task corpus consists of 600 PubMed abstracts annotated for over 17,000 events and was prepared as an extension of the MLEE Pyysalo et al. (2012a) corpus of 250 abstracts (c.f. Section A.1.1). The PC task corpus consists of 525 PubMed abstracts, chosen for the relevance to specific pathway reactions selected from SBML models registered in BioModels and PANTHER DB repositories Mi and Thomas (2009). The corpus was manually annotated for over 12,000 events on top of close to 16,000 entities.

Conversion The BioNLP'13 corpora are available from http://2013.bionlp-st.org/ in the same standoff format as the '09 and '11 corpora (Sections A.1.5 and A.1.6). As for these resources, we use the standard training, development and test set splits of each corpus and all of the physical entity annotations available for each dataset, and perform the conversion using the `standoff2conll` tool. Of the BioNLP'13 corpora only the CG task involved overlap between annotations in the source data; these were resolved in favour of keeping the shorter annotations, as for BioNLP'11 ID processing.

The conversion was highly accurate for all three of the BioNLP'13 corpora: the numbers of annotations in the converted data were 99.07%, 99.91%, and 99.95% of the numbers

[1] Option `-o keep-shorter` for `standoff2conll`

of annotations in the source for CG, GE, and PC respectively. Similarly, the fractions of annotated strings matching after conversion were 98.67%, 98.79%, and 99.80% (resp.).

A.1.8 CRAFT Corpus

The Colorado Richly Annotated Full Text (CRAFT) corpus Bada et al. (2012); Verspoor et al. (2012) consists of 67 full-text articles, over 790,000 Tokens, over 21,000 Sentences and approximately 140,000 concept annotations. It manually annotates all mentions of nearly all concepts from nine prominent biomedical ontologies and terminologies: Cell Type Ontology, Chemical Entities of Biological Interest ontology, NCBI Taxonomy, Protein Ontology, Sequence Ontology, Entrez Gene database entries, and the three sub-ontologies of the Gene Ontology. There was emphasis on journal articles that comprise the corpus being drawn from diverse biomedical disciplines and on them being completely annotated. We use the annotated physical entities from this corpus.

Conversion

The 67 publicly released articles of the CRAFT corpus are available in multiple formats from http://bionlp-corpora.sourceforge.net/CRAFT/. We split the data into 34 training, 11 development and 22 test documents and created a custom conversion for the corpus from the Knowtator format Ogren (2006). Of the resources considered in this study, the CRAFT term annotations represented the most challenges for use in sequence labelling: these are frequently overlapping, occasionally discontinuous, and associated with ontology identifiers (e.g. PR:000009758) rather than simple labels such as PROTEIN. To convert the corpus, we first excluded annotations not associated with physical entity types (biological process/molecular function, coreference, sections and typography). We then merged annotations associated with gene (ENTREZGENE) and protein (PR) identifiers, which frequently mark identical spans in the source data, into a single gene/gene-product type. We likewise merged those referencing ORGANISM and TAXONOMIC RANK vocabularies. We finally deduplicated the resulting annotations and resolved remaining overlapping and discontinuous entities with corpus-specific heuristics implemented in a custom tool available from https://github.com/spyysalo/knowtator2standoff/.

The resulting dataset contains 72.05% of the number of annotations in the physical entity-associated subsets of CRAFT (CHEBI, CL, ENTREZGENE, GO-CC, NCBITAXON, PR, and SO), with 69.76% of the annotated names in the source matching ones in the converted data. These numbers are by far the lowest among the corpora considered here. While most of the difference reflects fundamental limitations of the BIO representation, many decisions in

the conversion could reasonably be made in another way and our results on CRAFT should thus not be directly compared to others where a different conversion of the data has been used.

A.1.9 Ex-PTM Corpus

The Exhaustive Post-Translational Modifications corpus Pyysalo et al. (2011) was part of the BioNLP Shared Task 2011 and employed a similar creation methodology to that of the BioNLP11 EPI task corpus (c.f. Subsection A.1.6). It annotated 360 PubMed abstracts containing 76,806 words of which 4,698 were annotated as proteins. Though the more semantically complex PTM identification task used manual annotations, the Protein/Gene entity mentions were automatically tagged using the BANNER Leaman and Gonzalez (2008) named entity tagger trained on the GENETAG Tanabe et al. (2005) corpus. Abstracts containing fewer than five entities were removed and a randomly chosen subset of the remaining documents were annotated.

Conversion The Exhaustive PTM corpus is available from http://www.geniaproject.org/ in the standoff format used by the BioNLP corpora (Section A.1.5). Unlike the shared task resources, the Ex-PTM corpus does not come with a pre-defined development set, but only a split between training and test data; we thus split off 49 of the 196 test documents as a development set. Conversion of the single physical entity annotation type, PROTEIN, was again performed with `standoff2conll`. As the source data contained a small number of non-ASCII characters, the conversion tool was run with the `-a` option to map these to ASCII.

The conversion exactly preserves the number of annotations in the source data. However, as for the BioNLP'11 EPI corpus (Section A.1.6) with which the Ex-PTM corpus shares a domain and some development history, the fraction of original names matching the text of converted names is notably lower at 95.72%, reflecting comparatively frequent entity mention boundaries inside alphanumeric tokens.

A.1.10 JNLPBA Corpus

The Joint workshop on NLP in Biomedicine and its Applications corpus consists of 2,404 publication abstracts (approx. 22,400 sentences) and is annotated for mentions of five entity types: CELL LINE, CELL TYPE, DNA, RNA, and PROTEIN Kim et al. (2004). The corpus was derived from GENIA corpus entity annotations. It is now a standard point of reference for evaluating multi-class biomedical entity taggers and has served as training material for tools such as ABNER Settles (2005) and the GENIA Tagger.

Conversion The JNLPBA corpus is available from http://www.geniaproject.org/ and distributed in the CoNLL IOB format with a split into train and test subsets. To create the development set, we separated 200 of the 2000 documents from the training data. As format conversion was not required, the annotations match the original data exactly.

A.1.11 LINNAEUS Corpus

The LINNAEUS corpus Gerner et al. (2010) consists of 100 full-text documents from the PMCOA document set which were randomly selected. All mentions of species terms were manually annotated and normalized to the NCBI taxonomy IDs of the intended species.

Conversion The LINNAEUS corpus is available from http://linnaeus.sourceforge.net/ in a TAB-separated standoff format. The resource does not define training, development or test subsets. We converted the corpus into BioNLP shared task standoff format using a custom script available from https://github.com/spyysalo/linnaeus-corpus, split it into 50-, 17-, and 33-document training, development and test sets, and then converted these into the CoNLL format using `standoff2conll`. As a full-text corpus, LINNAEUS contains comparatively frequent non-ASCII characters, which were mapped to ASCII using the `standoff2conll -a` option.

The conversion was highly accurate, but due to sentence-splitting errors within entity mentions, the number of annotations in the converted data was larger by four (100.09%) than that in the source data. 99.77% of names in the original annotation matched names in the converted data.

A.1.12 NCBI Disease Corpus

The NCBI Disease corpus Doğan et al. (2014) consists of 793 PubMed abstracts fully annotated at the mention and concept level for disease mentions. The public release of the NCBI disease corpus contains 6,892 disease mentions, which are mapped to 790 unique disease concepts. Of these, 88% link to a MeSH identifier, while the rest contain an OMIM identifier. 91% of the mentions were linked to a single disease concept, while the rest are described as a combination of concepts.

Conversion The NCBI Disease corpus is available in a TAB-separated standoff format with a standard split into training, development and test subsets from http://www.ncbi.nlm.nih.gov/CBBresearch/Dogan/DISEASE/. We converted the corpus annotations to CoNLL format using tools available from https://github.com/spyysalo/ncbi-disease. The converted

number of annotations was 99.84% of the original number, with 99.81% of strings in the original annotations matching with converted data. The differences were mostly due to a duplicated document in the source data.

A.1.13 GENIA POS

The GENIA corpus is one of the most widely used resources for biomedical NLP and has a rich set of annotations including parts of speech, phrase structure syntax, entity mentions, and events Ohta et al. (2002). For this work we use the GENIA POS annotations, which cover 2000 PubMed abstracts (approx. 20,000 sentences).

Conversion We use the GENIA corpus v3.02 POS annotations that were used to train the GENIA tagger Tsuruoka et al. (2005), available from https://github.com/spyysalo/genia-pos.[2] We split off 210 of the 1790 training set documents into a development test. The data is distributed in a tagged-token format that could be straightforwardly recast into the CoNLL format, preserving both the tokenisation and the annotations of the original exactly.

A.2 Complete Results of MTL Effects

To determine the exact effect that each NER dataset had on every other one, the multi-task model described in the paper was used to train each NER dataset with every other one. That is, a Multi-output multi-task model was trained for each ordered combination of the datasets to give 15 x 14 models. The best results for each dataset was included in the paper, but the full set of all results could not be included for space considerations. They are added in Table 3.2.

Dataset	Scores
AnatEM	**BC2GM**: 80.63, **BC4CHEMD**: 77.72, **BC5CDR**: 80.85, **BioNLP09**: 80.99, **BioNLP11EPI**: 80.81, **BioNLP11ID**: 81.22, **BioNLP13CG**: 81.14, **BioNLP13GE**: 81.48, **BioNLP13PC**: 81.03, **CRAFT**: 80.03, **Ex-PTM**: 81.57, **JNLPBA**: 78.20, **Linnaeus**: 80.94, **NCBI-Disease**: 81.68*
BC2GM	**AnatEM**: 72.07, **BC4CHEMD**: 68.32, **BC5CDR**: 71.80, **BioNLP09**: 71.43, **BioNLP11EPI**: 71.95, **BioNLP11ID**: 71.56, **BioNLP13CG**: 71.68, **BioNLP13GE**: 72.17, **BioNLP13PC**: 72.04, **CRAFT**: 70.20, **Ex-PTM**: 72.21*,

[2]We are grateful to Yoshimasa Tsuruoka for providing this version of the corpus, which differs from that available from http://www.geniaproject.org/ most importantly in providing a train/test split.

Dataset	Scores
	JNLPBA: 69.35, **Linnaeus**: 71.64, **NCBI-Disease**: 71.84
BC4CHEMD	**AnatEM**: 79.58, **BC2GM**: 78.84, **BC5CDR**: 79.43, **BioNLP09**: 79.34 **BioNLP11EPI**: 79.91, **BioNLP11ID**: 79.35, **BioNLP13CG**: 78.98, **BioNLP13GE**: 80.31*, **BioNLP13PC**: 79.54, **CRAFT**: 78.19, **Ex-PTM**: 80.29, **JNLPBA**: 77.37, **Linnaeus**: 79.39, **NCBI-Disease**: 79.57
BC5CDR	**AnatEM**: 83.21, **BC2GM**: 82.54, **BC4CHEMD**: 81.45, **BioNLP09**: 83.18, **BioNLP11EPI**: 83.77*, **BioNLP11ID**: 83.38, **BioNLP13CG**: 83.66, **BioNLP13GE**: 83.54, **BioNLP13PC**: 83.58, **CRAFT**: 81.95, **Ex-PTM**: 83.03, **JNLPBA**: 81.10, **Linnaeus**: 83.28, **NCBI-Disease**: 83.72
BioNLP09	**AnatEM**: 83.24, **BC2GM**: 83.56, **BC4CHEMD**: 81.89, **BC5CDR**: 83.35, **BioNLP11EPI**: 84.14, **BioNLP11ID**: 83.50, **BioNLP13CG**: 83.68, **BioNLP13GE**: 84.16*, **BioNLP13PC**: 83.53, **CRAFT**: 82.97, **Ex-PTM**: 83.86, **JNLPBA**: 82.29, **Linnaeus**: 82.78, **NCBI-Disease**: 83.55
BioNLP11EPI	**AnatEM**: 76.62, **BC2GM**: 76.60, **BC4CHEMD**: 74.48, **BC5CDR**: 76.67, **BioNLP09**: 78.10*, **BioNLP11ID**: 76.86, **BioNLP13CG**: 76.97, **BioNLP13GE**: 77.49, **BioNLP13PC**: 77.14, **CRAFT**: 75.80, **Ex-PTM**: 77.99, **JNLPBA**: 74.87, **Linnaeus**: 76.62, **NCBI-Disease**: 76.51
BioNLP11ID	**AnatEM**: 81.43, **BC2GM**: 81.35, **BC4CHEMD**: 77.16, **BC5CDR**: 81.43, **BioNLP09**: 81.87, **BioNLP11EPI**: 81.76, **BioNLP13CG**: 81.90, **BioNLP13GE**: 82.26*, **BioNLP13PC**: 81.66, **CRAFT**: 80.36, **Ex-PTM**: 81.73, **JNLPBA**: 78.80, **Linnaeus**: 81.62, **NCBI-Disease**: 81.78
BioNLP13CG	**AnatEM**: 75.85, **BC2GM**: 73.94, **BC4CHEMD**: 68.73, **BC5CDR**: 76.05, **BioNLP09**: 75.41, **BioNLP11EPI**: 75.78, **BioNLP11ID**: 76.58, **BioNLP13GE**: 76.26, **BioNLP13PC**: 77.33*, **CRAFT**: 74.08, **Ex-PTM**: 77.16, **JNLPBA**: 70.46, **Linnaeus**: 75.09, **NCBI-Disease**: 75.72
BioNLP13GE	**AnatEM**: 74.05, **BC2GM**: 74.08, **BC4CHEMD**: 73.19, **BC5CDR**: 73.48, **BioNLP09**: 75.99, **BioNLP11EPI**: 76.09*, **BioNLP11ID**: 73.66, **BioNLP13CG**: 75.35, **BioNLP13PC**: 73.99, **CRAFT**: 75.46, **Ex-PTM**: 73.78, **JNLPBA**: 74.15, **Linnaeus**: 74.16, **NCBI-Disease**: 74.05
BioNLP13PC	**AnatEM**: 79.61, **BC2GM**: 77.78, **BC4CHEMD**: 75.72, **BC5CDR**: 79.79, **BioNLP09**: 79.08, **BioNLP11EPI**: 79.31, **BioNLP11ID**: 80.67, **BioNLP13CG**: 80.36, **BioNLP13GE**: 80.76, **CRAFT**: 77.66, **Ex-PTM**: 80.94*, **JNLPBA**: 78.73, **Linnaeus**: 78.60, **NCBI-Disease**: 79.55
CRAFT	**AnatEM**: 77.08, **BC2GM**: 76.97, **BC4CHEMD**: 73.61, **BC5CDR**: 77.97,

Dataset	Scores
	BioNLP09: 77.70, **BioNLP11EPI**: 77.61, **BioNLP11ID**: 78.10, **BioNLP13CG**: 77.30, **BioNLP13GE**: 78.48*, **BioNLP13PC**: 77.93, **Ex-PTM**: 78.36, **JNLPBA**: 74.86, **Linnaeus**: 77.38, **NCBI-Disease**: 77.43
Ex-PTM	**AnatEM**: 68.45, **BC2GM**: 68.35, **BC4CHEMD**: 60.33, **BC5CDR**: 69.46, **BioNLP09**: 72.00, **BioNLP11EPI**: 73.58*, **BioNLP11ID**: 69.58, **BioNLP13CG**: 68.82, **BioNLP13GE**: 70.07, **BioNLP13PC**: 70.36, **CRAFT**: 67.25, **JNLPBA**: 62.60, **Linnaeus**: 69.20, **NCBI-Disease**: 68.49
JNLPBA	**AnatEM**: 68.19, **BC2GM**: 68.20, **BC4CHEMD**: 66.49, **BC5CDR**: 68.77, **BioNLP09**: 68.11, **BioNLP11EPI**: 68.33, **BioNLP11ID**: 68.19, **BioNLP13CG**: 68.54, **BioNLP13GE**: 68.92*, **BioNLP13PC**: 68.84, **CRAFT**: 67.97, **Ex-PTM**: 68.84, **Linnaeus**: 68.18, **NCBI-Disease**: 68.51
Linnaeus	**AnatEM**: 83.23, **BC2GM**: 81.71, **BC4CHEMD**: 79.24, **BC5CDR**: 82.83, **BioNLP09**: 83.12, **BioNLP11EPI**: 82.20, **BioNLP11ID**: 81.77, **BioNLP13CG**: 80.47, **BioNLP13GE**: 82.81, **BioNLP13PC**: 82.68, **CRAFT**: 81.21, **Ex-PTM**: 82.37, **JNLPBA**: 77.06, **NCBI-Disease**: 83.63*
NCBI-Disease	**AnatEM**: 79.76, **BC2GM**: 78.40, **BC4CHEMD**: 75.16, **BC5CDR**: 79.98, **BioNLP09**: 78.97, **BioNLP11EPI**: 79.75, **BioNLP11ID**: 79.24, **BioNLP13CG**: 79.85, **BioNLP13GE**: 80.06, **BioNLP13PC**: 79.41, **CRAFT**: 76.96, **Ex-PTM**: 80.74*, **JNLPBA**: 74.84, **Linnaeus**: 79.21

Table A.2 Full Effects Results. (*: best score)

Dataset	Contents	Entity Counts
AnatEM (Pyysalo and Ananiadou, 2013)	Anatomy	13,701
BC2GM (Smith et al., 2008)	Gene/Protein	24,583
BC4CHEMD (Krallinger et al., 2015)	Chemical	84,310
BC5CDR (Wei et al., 2015)	Chemical, Disease	**Chemical:** 15,935; **Disease:** 12,852
BioNLP09 (Kim et al., 2008)	Gene/Protein	14,963
BioNLP11EPI (Pyysalo et al., 2012b)	Gene/Protein	15,811
BioNLP11ID (Pyysalo et al., 2012b)	4 NEs	**Gene/Protein:** 6,551; **Organism:** 3,471; **Chemical:** 973; **Regulon-operon:** 87
BioNLP13CG (Pyysalo et al., 2015)	16 NEs	**Gene/Protein:** 7,908; **Cell:** 3,492; **Cancer:** 2,582; **Chemical:** 2,270; **Organism:** 1,715; **Multi-tissue structure:** 857; **Tissue:** 587; **Cellular component:** 569; **Organ:** 421; **Organism substance:** 283; **Pathological formation:** 228; **Amino acid:** 135; **Immaterial anatomical entity:** 102; **Organism subdivision:** 98; **Anatomical system:** 41; **Developing anatomical structure:** 35
BioNLP13GE (Kim et al., 2013)	Gene/Protein	12,057
BioNLP13PC (Ohta et al., 2013)	4 NEs	**Gene/Protein:** 10,891; **Chemical:** 2,487; **Complex:** 1,502; **Cellular component:** 1,013
CRAFT (Bada et al., 2012)	6 NEs	**SO:** 18,974; **Gene/Protein:** 16,064; **Taxonomy:** 6,868; **Chemical:** 6,053; **CL:** 5,495; **GO-CC:** 4,180
Ex-PTM (Pyysalo et al., 2011)	Gene/Protein	4,698
JNLPBA (Kim et al., 2004)	5 NEs	**Gene/Protein:** 35,336; **DNA:** 10,589; **Cell Type:** 8,639; **Cell Line:** 4,330; **RNA:** 1,069
Linnaeus (Gerner et al., 2010)	Species	4,263
NCBI-Disease (Doğan et al., 2014)	Disease	6,881
GENIA-PoS (Ohta et al., 2002)	PoS-Tags	N/A

Table A.1 The datasets and details of their annotations

Appendix B

Neural Biomedical Link Prediction

B.1 Introduction

This is the appendix for Chapter 4. It contains additional results and analysis.

For SDNE, two implementations were tried: the one created by the authors Wang et al. (2016) and one created by Goyal and Ferrara (2018b). We used the parameters from Goyal and Ferrara (2018b) because our attempted hyper-parameters did not give good results and, though we contacted both sets of authors, only they responded to our request for the hyper-parameters used in their experiments.

B.2 Additional Results and Discussion

In the result tables, the number in **bold** represent the best score for a particular metric. The difference between the best and scores with an asterisk (*) are not statistically significant.

B.2.1 MATADOR

These results are in Table 4.5. The additional result is that SDNE is much worse than the other approaches for this dataset. This may be due to the fact that it is the deepest of all the neural network approaches and so required more data to train properly. In the main paper, we already attribute the relatively poor performance of the deep learning models compared to the baselines to the small size of this dataset - that argument would hold even more so for SDNE.

Note also that LINE embeddings combined with Hadamard were on par with the best performer for precision at k.

Method	Node Combination	AUC (ROC)	AUC (PR)	MAP	Avg. R-prec	Prec @ k
Deep-Walk	Average	95.93 ± .003	95.82 ± .005	89.81 ± .003	86.86 ± .003	98.77 ± .004*
	Concat	94.97 ± .004	94.83 ± .003	88.30 ± .000	84.63 ± .001	98.34 ± .002*
	Hadamard	90.21 ± .003	91.55 ± .004	86.65 ± .01	82.59 ± .01	97.56 ± .005
	W-L1	80.45 ± .01	82.74 ± .01	69.27 ± .006	62.56 ± .000	93.74 ± .02
	W-L2	85.67 ± .001	88.12 ± .004	77.31 ± .004	71.57 ± .005	97.44 ± .004
LINE	Average	80.63 ± .01	81.30 ± .006	67.74 ± .02	61.04 ± .03	91.65 ± .009
	Concat	81.16 ± .01	81.82 ± .007	68.53 ± .02	61.42 ± .02	92.00 ± .009
	Hadamard	89.11 ± .008	90.37 ± .006	83.45 ± .01	77.47 ± .02	98.00 ± .003
	W-L1	70.76 ± .02	79.32 ± .009	73.86 ± .007	66.15 ± .006	98.02 ± .009*
	W-L2	69.52 ± .02	76.37 ± .01	70.94 ± .003	63.33 ± .001	92.38 ± .02
node-2vec	Average	78.38 ± .02	78.75 ± .02	66.42 ± .02	59.32 ± .02	88.67 ± .01
	Concat	77.62 ± .03	77.54 ± .03	65.44 ± .02	58.40 ± .02	87.25 ± .03
	Hadamard	84.74 ± .03	85.12 ± .02	82.34 ± .02	76.88 ± .02	93.71 ± .02
	W-L1	75.38 ± .05	74.98 ± .05	69.32 ± .03	62.08 ± .04	83.94 ± .05
	W-L2	74.31 ± .05	74.57 ± .05	69.56 ± .03	62.48 ± .04	84.62 ± .05
SDNE	Average	55.77 ± .02	55.22 ± .03	54.81 ± .02	47.21 ± .02	57.56 ± .05
	Concat	54.88 ± .01	54.17 ± .01	53.37 ± .01	46.14 ± .01	56.41 ± .02
	Hadamard	53.12 ± .02	52.20 ± .02	51.81 ± .01	47.85 ± .07	52.84 ± .03
	W-L1	54.35 ± .01	53.44 ± .01	50.06 ± .06	45.56 ± .03	54.93 ± .03
	W-L2	52.60 ± .01	51.34 ± .01	50.67 ± .01	43.41 ± .01	50.44 ± .01
AA	N/A	91.97 ± .001	88.40 ± .002	87.16 ± .001	85.06 ± .003	86.87 ± .006
CN	N/A	**97.27 ± .002**	97.04 ± .003*	**95.47 ± .002**	**94.64 ± .002**	98.74 ± .004*
JC	N/A	97.23 ± .002*	**97.10 ± .001**	94.72 ± .002	92.29 ± .002	**98.96 ± .002**

Table B.1 MATADOR random-slice results

B.2.2 BioGRID

The randomly sliced experiments on this dataset are in Table 4.6 and the time-sliced experiments are in Table B.3.

Random-Slice

Node2vec embeddings combined with Hadamard were on par with the best performer for precision at k.

Time Slice

Section 3.1 of the paper explains why it is more difficult to perform link prediction in the time-slice setting. To recap: first, new nodes can be introduced to the graph at later time

periods which will present little or no information to the link predictor to use as they will have no links to other nodes in the time period which the predictor uses to make predictions. Second, in evolving graphs, the easier links tend to form first and more difficult ones later, so the edges to be predicted in later time periods tend to be more difficult.

Method	**Node Combination**	**AUC (ROC)**	**AUC (PR)**	**MAP**	**Avg. R-prec**	**Prec @ *k***
DeepWalk	Average	97.69 ± .000	97.62 ± .001	79.24 ± .003	73.86 ± .003	99.30 ± .001
	Concat	97.74 ± .001	97.65 ± .002	82.48 ± .006	77.70 ± .006	99.18 ± .002
	H'mard	95.76 ± .001	96.54 ± .001	79.63 ± .001	74.87 ± .001	99.25 ± .001
	W-L1	79.17 ± .004	80.57 ± .004	51.96 ± .008	46.50 ± .009	91.71 ± .005
	W-L2	79.73 ± .002	81.08 ± .001	52.81 ± .002	47.39 ± .003	92.12 ± .001
LINE	Average	98.10 ± .00*	97.80 ± .00*	83.13 ± .02*	78.22 ± .02*	99.54 ± .00*
	Concat	98.08 ± .000	97.76 ± .000	82.94 ± .004	78.04 ± .009	99.29 ± .001
	H'mard	94.45 ± .002	95.35 ± .002	80.17 ± .001	75.17 ± .01	99.30 ± .002
	W-L1	92.41 ± .006	92.06 ± .006	70.88 ± .009	65.21 ± .008	97.07 ± .003
	W-L2	91.80 ± .006	91.55 ± .006	71.80 ± .003	66.39 ± .005	96.56 ± .005
node2vec	Average	98.32 ± .00*	97.97 ± .03*	85.70 ± .01*	81.17 ± .01*	99.38 ± .00*
	Concat	**98.51 ± .001**	**98.26 ± .03**	**86.49 ± .009**	**81.84 ± .009**	99.49 ± .00*
	H'mard	97.19 ± .001	97.17 ± .03	81.53 ± .01	76.54 ± .01	99.33 ± .00*
	W-L1	92.02 ± .007	92.30 ± .03	64.24 ± .01	59.45 ± .008	97.45 ± .003
	W-L2	93.07 ± .003	93.01 ± .03	67.11 ± .007	61.94 ± .005	97.47 ± .005
AA	N/A	86.10 ± .000	90.75 ± .001	70.97 ± .001	57.65 ± .001	96.13 ± .001
CN	N/A	91.20 ± .000	94.96 ± .000	75.72 ± .000	69.81 ± .003	**99.64 ± .000**
JI	N/A	90.80 ± .000	93.95 ± .000	73.93 ± .001	68.79 ± .001	98.59 ± .000

Table B.2 BioGRID random-slice results

As expected, the majority of the approaches performed worse in all metrics than the randomly sliced experiments with this dataset. However there were some exceptions. DeepWalk embeddings combined by Weighted-L1 and L2, node2vec embeddings combined with Weighted-L1 and all baselines recorded better performance for MAP. DeepWalk embeddings combined by Weighted-L1 and L2, node2vec embeddings combined with Weighted-L1 and Adamic-Adar recorded better performance for averaged R-precision. Adamic-Adar also recorded increased performance for precision at k. There are several possible contributing factors here.

For MAP and averaged R-precision, if a particular node has no positives it is removed from the calculations as these metrics are only concerned with predicted true positives. In the time-sliced data, there are a much higher percentage of nodes which have no true positives in the test slice than is the case with randomly-sliced data. These nodes are also likely to have a

Method	Node Combination	AUC (ROC)	AUC (PR)	MAP	Avg. R-prec	Prec @ k
DeepWalk	Average	89.40 ± .009	90.10 ± .01	68.94 ± .001	63.30 ± .001	97.25 ± .001*
	Concat	92.12 ± .004	92.78 ± .003	71.61 ± .002	65.96 ± .002	98.04 ± .002
	Hadamard	89.03 ± .004	91.39 ± .004	66.28 ± .002	60.34 ± .003	98.31 ± .003
	W-L1	69.75 ± .02	67.43 ± .01	59.74 ± .006	54.61 ± .006	73.26 ± .006
	W-L2	72.11 ± .01	69.33 ± .006	59.84 ± .004	54.51 ± .005	75.02 ± .005
LINE	Average	91.86 ± .006	92.31 ± .006	72.85 ± .002	67.76 ± .002	97.40 ± .002
	Concat	93.55 ± .003	93.74 ± .002	73.60 ± .002	68.57 ± .002	97.90 ± .002
	Hadamard	77.70 ± .02	82.51 ± .01	67.78 ± .004	61.33 ± .005	96.05 ± .005
	W-L1	82.36 ± .007	81.32 ± .009	66.66 ± .004	60.93 ± .005	88.54 ± .005
	W-L2	79.79 ± .03	78.82 ± .02	66.53 ± .002	60.75 ± .004	86.76 ± .004
node2vec	Average	**95.25 ± .002**	**95.43 ± .004**	74.91 ± .001	**70.39 ± .001**	98.26 ± .001
	Concat	93.66 ± .002	94.66 ± .004*	73.48 ± .002	68.77 ± .002	98.40 ± .002*
	Hadamard	93.94 ± .002	94.02 ± .009*	71.81 ± .003	66.57 ± .003	97.59 ± .003*
	W-L1	89.06 ± .002	88.70 ± .004	66.17 ± .005	61.20 ± .004	93.86 ± .004
	W-L2	88.81 ± .003	88.43 ± .006	66.09 ± .01	61.02 ± .01	93.54 ± .01
AA	N/A	77.46 ± .000	87.69 ± .000	74.84 ± .000	61.39 ± .001	98.10 ± .000
CN	N/A	85.07 ± .000	91.81 ± .000	**76.20 ± .001**	67.73 ± .004	**99.38 ± .000**
JC	N/A	84.74 ± .000	90.20 ± .001	75.60 ± .001	67.49 ± .000	97.45 ± .001

Table B.3 BioGRID time-slice results

small amount of links and are thus difficult nodes to perform well on, so it is not surprising that the approaches which performed poorest on the randomly-sliced version of this dataset benefited from having less and easier nodes in the evaluation. The poor embeddings created for this setting as explained above would contribute to decreased performance for the other methods but as all combination methods use the same embeddings, there is something about the DeepWalk embeddings combined with Weighted L1 and L2 which help in this setting.

Node2vec embeddings combined with Hadamard had performance that was not significantly worse than the best for AUPRC and precision at k.

B.2.3 PubTator

The randomly sliced experiments on this dataset can be seen in Table 4.7 and the time-sliced experiments can be seen in Table B.5.

Random-Slice

Nothing much to add here except to note that Common Neighbours outperformed the lower neural network performers (Hadamard, Weighted-L1 and Weighted-L2) for most metrics.

Time Slice

As with the BioGRID data, the majority of the approaches performed worse in this setting than the random-sliced one, and there were again some exceptions. DeepWalk embeddings combined by Weighted-L1 and L2 had better performance in all metrics and Adamic-Adar again recorded increased performance for precision at k. Similar explanations hold for this situation as well. In this case only the DeepWalk vectors were better and they were better in all metrics and the previous explanations pertained only to the node-level metrics. These results provide strong indication that DeepWalk embeddings combined with Weighted-L1 and Weighted-L2 perform better in the time sliced setting than the random slice one, but their performances are still significantly worse than the best performers in these settings.

Method	**Node Combination**	**AUC (ROC)**	**AUC (PR)**	**MAP**	**Avg. R-prec**	**Prec @ *k***
Deep-Walk	Average	98.85 ± .03	99.01 ± .02	83.67 ± .12	75.97 ± .28	99.93* ± .006
	Concat	**99.20 ± .006**	**99.30 ± .006**	**91.01 ± .16**	85.46 ± .20	99.94* ± .006
	Hadamard	98.44 ± .06	98.68 ± .03	84.67 ± .36	77.84 ± .31	99.88 ± .01
	W-L1	88.96 ± .40	89.63 ± .36	60.76 ± 1.7	51.21 ± 1.5	97.64 ± .16
	W-L2	89.25 ± .01	89.90 ± .07	62.10 ± .36	52.57 ± .40	97.67 ± .16
LINE	Average	99.10 ± .09*	99.23 ± .08*	90.36 ± .82*	84.56 ± 1.0	**99.97 ± .03**
	Concat	99.13 ± .02	99.24 ± .02	90.07 ± .34	84.03 ± .48	99.95 ± .006*
	Hadamard	98.30 ± .04	98.49 ± .05	86.40 ± .69	79.28 ± .87	99.90 ± .006
	W-L1	93.93 ± .10	94.16 ± .10	78.25 ± .94	69.48 ± 1.1	98.97 ± .13
	W-L2	94.23 ± .11	94.51 ± .02	77.97 ± .96	69.00 ± 1.2	99.13 ± .06
node-2vec	Average	98.71 ± .05	98.90 ± .04	82.98 ± .58	75.29 ± .72	99.94 ± .006*
	Concat	99.16 ± .03*	99.21 ± .02	88.94 ± .29	82.14 ± .30	99.92 ± .0*
	Hadamard	98.81 ± .03	98.91 ± .02	86.40 ± .22	79.07 ± .27	99.87 ± .006
	W-L1	88.07 ± .03	87.28 ± .11	87.28 ± 1.4	48.95 ± 1.4	94.08 ± .16
	W-L2	88.85 ± .07	88.26 ± .02	88.26 ± .74	50.72 ± .69	94.90 ± .13
AA	N/A	92.92 ± .03	84.56 ± .04	56.48 ± .16	66.38 ± .13	83.33 ± .02
CN	N/A	98.40 ± .01	98.28 ± .01	79.84 ± .19	**87.10 ± .16**	99.94 ± .00*
JI	N/A	92.36 ± .02	87.59 ± .03	65.44 ± .05	59.74 ± .04	91.21 ± .01

Table B.4 PubTator random-slice results

Method	Node Combination	AUC (ROC)	AUC (PR)	MAP	Avg. R-prec	Prec @ *k*
Deep-Walk	Average	93.86 ± .00*	95.51 ± .00*	70.78 ± .00*	62.16 ± .00*	**99.89 ± .000**
	Concat	**93.99 ± .002**	**95.70 ± .001**	**71.11 ± .003**	**62.65 ± .003**	**99.89 ± .00**
	Hadamard	87.23 ± .002	91.33 ± .001	54.72 ± .002	46.22 ± .002	99.70 ± .001
	W-L1	92.06 ± .001	93.23 ± .000	66.47 ± .001	57.29 ± .001	98.77 ± .000
	W-L2	91.81 ± .002	93.06 ± .002	65.89 ± .003	56.66 ± .004	98.76 ± .000
LINE	Average	88.68 ± .03*	92.27 ± .02*	55.61 ± .09*	46.41 ± .09*	**99.89 ± .000**
	Concat	90.32 ± .005	93.01 ± .002	62.51 ± .02	53.21 ± .02	**99.89 ± .001**
	Hadamard	87.09 ± .007	89.98 ± .005	51.97 ± .01	42.43 ± .01	99.10 ± .003
	W-L1	83.58 ± .001	86.55 ± .004	47.71 ± .003	38.11 ± .002	97.26 ± .007
	W-L2	82.81 ± .003	85.79 ± .003	47.07 ± .005	37.49 ± .004	96.78 ± .006
node-2vec	Average	88.40 ± .003	92.07 ± .002	55.72 ± .003	46.48 ± .004	99.87 ± .000
	Concat	88.13 ± .001	91.83 ± .000	53.24 ± .002	43.69 ± .004	99.84 ± .000
	Hadamard	85.24 ± .001	90.63 ± .001	47.76 ± .003	38.84 ± .003	99.81 ± .00*
	W-L1	84.68 ± .003	89.08 ± .001	44.69 ± .003	35.34 ± .003	98.57 ± .00
	W-L2	84.48 ± .001	89.12 ± .000	44.68 ± .001	35.49 ± .000	98.67 ± .000
AA	N/A	85.10 ± .000	80.24 ± .000	35.49 ± .000	40.13 ± .000	90.56 ± .001
CN	N/A	88.37 ± .000	88.83 ± .000	43.67 ± .000	46.59 ± .000	99.84 ± .000
JI	N/A	86.08 ± .000	83.52 ± .000	38.66 ± .000	38.75 ± .001	94.27 ± .000

Table B.5 PubTator time-slice results

B.3 Additional *K* Values for Precision at *k*

The main manuscript lists results for precision at k when k=30% of all positives. Here we add additional results fro k= 10, 20 and 30.

Method	**Node Combination**	**P@10**	**P@20**	**P@40**
Deep-	Average	99.47	99.04	98.26
Walk	Concat	99.65	98.87	98.22
	Hadamard	98.61	98.26	98.22
	W-L1	98.61	98.87	91.66
	W-L2	98.78	98.87	96.61
LINE	Average	93.03	91.98	88.51
	Concat	93.73	93.12	89.60
	Hadamard	92.33	90.33	86.55
	W-L1	98.26	98.34	98.12
	W-L2	95.12	93.12	89.60
node-	Average	89.91	89.40	84.75
2vec	Concat	92.35	89.57	86.62
	Hadamard	95.65	94.01	90.36
	W-L1	92.17	91.49	86.53
	W-L2	94.43	92.79	87.92
SDNE	Average	57.04	54.96	54.00
	Concat	55.83	53.30	52.00
	Hadamard	55.83	55.91	53.91
	W-L1	53.22	53.57	53.26
	W-L2	50.96	48.61	49.61
AA	N/A	61.32	66.18	73.88
CN	N/A	97.49	98.36	97.10
JC	N/A	97.10	98.07	97.54

Table B.6 MATADOR additional P@*K* results

Method	Node Combination	P@10	P@20	P@40
Deep-	Average	99.61	99.50	99.23
Walk	Concat	99.69	99.59	99.33
	Hadamard	99.42	99.39	99.25
	W-L1	97.68	94.00	87.12
	W-L2	97.29	94.74	89.30
LINE	Average	99.48	99.37	99.14
	Concat	99.63	99.57	99.27
	Hadamard	99.56	99.37	98.94
	W-L1	99.11	98.36	96.81
	W-L2	98.90	97.59	95.90
node-	Average	99.61	99.54	99.25
2vec	Concat	99.62	99.53	99.29
	Hadamard	99.31	99.28	99.02
	W-L1	98.24	97.97	97.35
	W-L2	98.11	97.70	96.90
AA	N/A	93.52	94.83	96.47
CN	N/A	99.79	99.72	99.56
JC	N/A	98.21	98.49	98.45

Table B.7 BioGRID additional P@K results

Method	Node Combination	P@10	P@20	P@40
Deep-	Average	99.12	98.67	97.36
Walk	Hadamard	97.99	97.22	95.36
	W-L1	98.48	97.79	96.39
	W-L2	98.55	97.94	96.59
LINE	Average	99.08	98.59	97.16
	Hadamard	95.62	94.45	92.06
	W-L1	96.84	94.89	90.48
	W-L2	96.87	95.32	91.14
AA	N/A	85.62	88.39	92.17
CN	N/A	99.10	98.60	96.92
JC	N/A	82.32	84.89	86.67

Table B.8 PubTator additional P@K results

Appendix C

Towards integration – Comparison with a Real-world LBD System

C.1 Introduction

This Appendix contains supplementary information for Chapter 5. It contains additional results and analysis which were left out of the main Chapter.

C.2 Formal Definitions of Evaluation Metrics

1. **Mean Average Precision (MAP):** Given a ranked list of predicted terms (C) relevant to a particular query (A) term, we can calculate the precision after each true positive. The average of these values gives the average precision for that query. This done over all queries gives a single value measure which weights all queries (difficult or easy) equally.
   $MAP = \frac{\Sigma_i AP(i)}{|V|}$,
   where $|V|$ = number of queries, AP(i) = $\Sigma_n(R_n - R_{n-1})P_n$ and P_n and R_n are the Precision and Recall at the n^{th} threshold for the i^{th} query.

2. **Mean Reciprocal Rank (MRR):**
   $\Sigma_i \frac{1}{rank(i)}$,
   where $rank(i)$ = absolute rank for the i^{th} query.

3. **Averaged R(elevant)-Precision:** Similar to MAP but instead of calculating the precision after each positive term (gold C) in the list of results for a given query, precision is only calculated with the top R results. R is determined by how many true positives exist

for the query. The main difference from MAP is that this metric does not consider the remainder of the ranked list outside of the length of the top R. This also gives a single value measure which weights all queries equally. This metric is similar to precision at k except that instead of having a fixed k, it changes based on the amount of positives each node has so that a query with less than k positives is not unfairly penalised and a query with a lot more positives than k is not easier for the approach to perform well at.

$$AveragedR - precision = \frac{\Sigma_i Pr@R(i)}{|V|},$$

where $|V|$ = number of nodes, $Pr@R(i)$ = precision at R for the i^{th} node with R positives.

C.3 Other Neural Network Hyperparameters

LINE: learning rate = 0.025, number of negative samples = 5 and total number of samples = 1 billion. According to Tang et al. (2015), LINE performs best when it is run twice to obtain first- and second-order proximity embeddings which are concatenated and L2 normalized. I follow their recommendations. For each order I created half the number of dimensions as needed so that when they were concatenated, the final result had the appropriate number.

C.4 Results

The results of the neural approaches are means of the means which were calculated over 5 runs. The standard deviations reported are of the mean ranks. The results of the baselines are means of the method across all relevant cases and the standard deviations are those over those ranks. The best rank is in **boldface** type. We sought to determine what methods gave the lowest mean ranks and lowest variance (measured by standard deviation). Where possible, we use results from Pyysalo et al. (2018).

Wherever there are models that do not use aggregators or accumulators, the results are simply placed in the first column - this is merely for convenience, the column headers *would not* apply to such models. The best for a particular approach is underlined while the best of all approaches is in **bold**.

There were some experiments which produced ties with the gold which were of an amount to make them useless for real-world use. We defined that number as 10; methods which produced more than 10 ties with the gold are reported with a '*' instead of their performance.

C.4.1 Cancer Discoveries and Swanson Cases

Results for Closed Discovery performed on the 5 Cancer discovery cases on which LION was originally evaluated are in Tables C.1 and C.2.

Results for Open Discovery performed on the 5 Cancer Discovery cases on which LION was evaluated as reported in the paper. Means are in Table C.3 and medians are in Table C.4.

Results for Open Discovery performed on the 5 Swanson cases on which LION was evaluated. Means are in Table C.5 and medians are in Table C.6.

Results for Open Discovery performed on the 5 Cancer and 5 Swanson cases on which LION was evaluated. Means are in Table C.7 and medians are in Table C.8.

Approach	Min	Avg	Max
NPMI	278.2	272.6	282.0
SCP	252.2	285	298.6
χ^2	268.2	258.0	269.8
***t*-test**	262.0	246.8	260.8
LLR	266.0	246.4	264.0
Jaccard	214.8	258.8	281.6
Count	233.2	249.6	245.2
Doc-count	236.8	224.4	222.2
CD-1-A	112.9	86.3	97.2
CD-1-C	151.2	94.5	89.7
CD-1-H	357.2	251.3	287.0
CD-1-W1	228.7	195.8	189.0
CD-1-W2	614.3	482.9	565.2
CD-2-A	86.9	-	-
CD-2-C	**48.7**	-	-
CD-2-H	143.1	-	-
CD-2-W1	402.6	-	-
CD-2-W2	63.8	-	-

Table C.1 Mean Ranks for Closed Discovery on the Cancer Discovery Cases

C.4.2 Published Interactions: BioGRID

The results of the BioGRID experiments are in the following tables. Each table is dedicated to a single metric: Mean Rank (MR), Mean Reciprocal Rank (MRR), Mean Average Precision (MAP) and Mean Relevance-precision (R-precision).

Due to rounding, some scores seem equal in the tables but are not. Where this occurs and involves a best performer, the unrounded number was used to break the ties.

Approach	Min	Avg	Max
NPMI	86.0	119.0	170.0
SCP	70.0	196.0	299.0
χ^2	74.0	196.0	270.0
***t*-test**	56.0	136.0	261.0
LLR	65.0	163.0	264.0
Jaccard	81.0	213.0	282.0
Count	245.0	181.0	245.0
Doc-count	231.0	169.0	222.0
CD-1-A	96.0	93.8	89.4
CD-1-C	158.6	**36.4**	38.8
CD-1-H	282.8	176.0	238.8
CD-1-W1	109.4	158.4	114.8
CD-1-W2	300.2	240.0	256.0
CD-2-A	52.4	-	-
CD-2-C	42.0	-	-
CD-2-H	62.2	-	-
CD-2-W1	180.6	-	-
CD-2-W2	48.8	-	-

Table C.2 Median Ranks for Closed Discovery on the Cancer Discovery Cases

C.5 Additional Analyses

The existing approaches performed much better on mean rank for open discovery than they did on closed discovery, so there was more room for improvement there. This lower baseline explains to some degree why the performance improvements were more pronounced for closed discovery (Table C.1).

The difference between mean and median as average shows across the various cancer and Swanson discovery cases: with the exception of open discovery on only the Cancer Discovery cases (Tables C.3 and C.4), the best performer for mean and median were different.

A conclusion to be drawn from all the results tables is that although the best neural network-based approaches performed the best, simply using neural networks is not sufficient to produce the best results as there are several instances where the best existing approaches outperformed some neural approaches.

Approach	**Min**		**Avg**		**Max**	
	Sum	**Max**	**Sum**	**Max**	**Sum**	**Max**
NPMI	73,670.4	14,658.8	310.2	11,354.6	60.2	3479.2
SCP	244.8	2,358.4	553.8	1,408.4	556.0	1,305.4
χ^2	37,387.4	2,971.6	603.4	1,521.2	601.4	1,469.6
***t*-test**	118,606.8	465.6	73,657.2	559.2	126.0	825.0
LLR	73,715.0	649.4	253.0	1,011.8	280.4	1,870.8
Jaccard	89.2	1,741.8	121.2	952.6	136.2	1,186.0
Count	367.4	2,063.6	412.6	1,483.6	421.0	875.8
Doc-count	394.4	2,141.8	472.6	1,249.2	490.6	2,071.2
OD-1-A	218.3	*	239.1	2,098.0	264.2	*
OD-1-C	93.4	*	123.2	37,248.0	156.9	*
OD-1-H	257.9	4,762.6	270.6	7,820.9	280.6	*
OD-1-W1	212.2	14,932.1	225.1	23,456.7	236.7	*
OD-1-W2	247.8	8,777.7	281.48	20,546.9	311.9	*
OD-2-A	127.9	-	-	-	-	-
OD-2-C	95,207.6	-	-	-	-	-
OD-2-H	**31.1**	-	-	-	-	-
OD-2-W1	57,226.2	-	-	-	-	-
OD-2-W2	582.9	-	-	-	-	-

Table C.3 Mean Ranks for Open Discovery on the Cancer Discovery Cases

Approach	Min		Avg		Max	
	Sum	**Max**	**Sum**	**Max**	**Sum**	**Max**
NPMI	98,698.0	15,476.0	121.0	5,897.0	36.0	2,268.0
SCP	276.0	926.0	400.0	1,176.0	399.0	727.0
χ^2	547.0	3,582.0	402.0	1,159.0	402.0	1,159.0
***t*-test**	118,751.0	63.0	98,406.0	325.0	125.0	176.0
LLR	98,677.0	187.0	344.0	646.0	319.0	645.0
Jaccard	29.0	1,089.0	78.0	962.0	93.0	1,122.0
Count	15.0	1,005.0	55.0	52.0	62.0	54.0
Doc-count	23.0	738.0	72.0	68.0	74.0	68.0
OD-1-A	26.8	*	38.6	1,212.6	48.6	*
OD-1-C	31.4	*	32.0	30,573.2	34.4	*
OD-1-H	46.2	1,750.3	46.6	8,120.4	49.4	*
OD-1-W1	28.6	8,905.0	33.4	21,335.2	39.2	*
OD-1-W2	43.8	8,370.2	49.0	18,442.8	55.2	*
OD-2-A	16.3	-	-	-	-	-
OD-2-C	98,148.2	-	-	-	-	-
OD-2-H	**12.2**	-	-	-	-	-
OD-2-W1	37,268.6	-	-	-	-	-
OD-2-W2	147.0	-	-	-	-	-

Table C.4 Median Ranks for Open Discovery on the Cancer Discovery Cases

Approach	**Min**		**Avg**		**Max**	
	Sum	**Max**	**Sum**	**Max**	**Sum**	**Max**
NPMI	39,481.0	12,805.6	27,041.8	13,290.0	4480.4	10,568.6
SCP	4,498.8	7,666.0	5,154.8	3,024.0	5,174.8	2,700.4
χ^2	37,873.6	10,402.8	5,182.6	4,702.8	5,319.6	3,803.2
***t*-test**	46,240.0	7,076.2	37,344.2	7,989.2	3,956.4	6,756.2
LLR	37,440.8	6,761.6	3,286.6	2,663.0	4,367.4	2,691.0
Jaccard	3,179.6	3,629.2	4,342.8	4,105.4	4,455.2	3,878.8
Count	3,484.2	2,882.2	4,242.0	2,216.0	4,265.2	5,364.4
Doc-count	3,470.8	2,871.0	4,229.6	2,199.8	4,255.6	5,365.2
OD-1-A	3,643.0	6,468.8	3,726.7	7,405.2	3,805.3	*
OD-1-C	3,721.4	11,229.8	3,757.4	16,325.9	3,788.6	*
OD-1-H	3,558.3	*	3,618.0	5,427.8	3,666.5	*
OD-1-W1	3,752.8	*	3,928.6	12,814.2	4,058.1	*
OD-1-W2	3,746.7	10,100.4	4,091.0	12,183.3	4,345.4	*
OD-2-A	6,859.0	-	-	-	-	-
OD-2-C	38,639.0	-	-	-	-	-
OD-2-H	**1,013.4**	-	-	-	-	-
OD-2-W1	29,960.9	-	-	-	-	-
OD-2-W2	14,697.4	-	-	-	-	-

Table C.5 Mean Ranks for Open Discovery on the Swanson Discovery Cases

Approach	Min		Avg		Max	
	Sum	**Max**	**Sum**	**Max**	**Sum**	**Max**
NPMI	41,837.0	8,869.0	16,714.0	9715.0	74.0	5,545.0
SCP	124.0	427.0	154.0	250.0	154.0	250.0
χ^2	37,827.0	7,820.0	156.0	263.0	155.0	263.0
***t*-test**	40,103.0	1,808.0	37,368.0	116.0	5.0	105.0
LLR	37,820.0	3,404.0	9.0	45.0	10.0	43.0
Jaccard	6.0	1,075.0	6.0	237.0	9.0	240.0
Count	8.0	43.0	20.0	29.0	21.0	261.0
Doc-count	7.0	21.0	20.0	31.0	21.0	237.0
OD-1-A	18.4	4,852.3	16.2	6,776.2	18.6	*
OD-1-C	**4.0**	1,917.8	9.6	6,933.0	16.4	*
OD-1-H	19.2	*	14.2	6,173.2	13.4	*
OD-1-W1	17.6	*	19.8	1,907.2	20.4	*
OD-1-W2	25.0	2,570.6	22.6	2,546.6	21.8	*
OD-2-A	605.4	-	-	-	-	-
OD-2-C	37,783.8	-	-	-	-	-
OD-2-H	17.6	-	-	-	-	-
OD-2-W1	44,254.0	-	-	-	-	-
OD-2-W2	49.6	-	-	-	-	-

Table C.6 Median Ranks for Open Discovery on the Swanson Discovery Cases

Approach	**Min**		**Avg**		**Max**	
	Sum	**Max**	**Sum**	**Max**	**Sum**	**Max**
NPMI	56,575.7	13,732.2	13,676	12,322.3	2,270.3	7,023.9
SCP	2,371.8	5,012.2	2,854.3	2,216.2	2,865.4	2,002.9
χ^2	37,630.5	6,687.2	2,893.0	3,112.0	2,960.5	2,636.4
***t*-test**	82,423.4	3,770.9	55,500.7	4,274.2	2,041.2	3,790.6
LLR	55,577.9	3,705.5	1,769.8	1,837.4	2,323.9	2,280.9
Jaccard	1634.4	2685.5	2,232.0	2,529.0	2,295.7	2,532.4
Count	1,925.8	2,472.9	2,327.3	1,849.8	2,343.1	3,120.1
Doc-count	1,932.0	2,506.4	2,351.1	1,724.5	2,373.1	3,718.2
OD-1-A	1,930.7	*	1,982.9	4,751.6	2,034.8	*
OD-1-C	1,907.42	*	1,940.3	26,786.9	1,972.8	*
OD-1-H	1,908.08	*	1,944.28	6,624.36	1,973.5	*
OD-1-W1	1,982.5	*	2,076.86	18,135.42	2,147.4	*
OD-1-W2	1,997.3	9,439.0	2,186.2	16,365.1	2,328.7	*
OD-2-A	3,493.5	-	-	-	-	-
OD-2-C	66,923.3	-	-	-	-	-
OD-2-H	**522.2**	-	-	-	-	-
OD-2-W1	43,593.5	-	-	-	-	-
OD-2-W2	7640.2	-	-	-	-	-

Table C.7 Mean Ranks for Open Discovery on the all Cases

Approach	Min		Avg		Max	
	Sum	**Max**	**Sum**	**Max**	**Sum**	**Max**
NPMI	50,347.0	10,624.0	698.0	9,472.0	55.0	3,630.0
SCP	200.0	758.5	370.5	9,472.0	371.0	630.0
χ^2	35,808.5	3,712.5	379.5	958.0	380.5	873.5
***t*-test**	78,806.0	344.5	48,107	220.5	43.5	169.5
LLR	48,337.0	569.0	44.5	420.0	46.5	540.5
Jaccard	21.0	1,082.0	46.5	610.5	57.5	849.0
Count	**11.5**	285.0	46.5	610.5	57.5	849.0
Doc-count	12.5	237.0	44.5	60.0	47.5	152.5
OD-1-A	25.2	*	29.2	3,850.4	32.7	*
OD-1-C	18.2	*	21.8	23,334.8	26.6	*
OD-1-H	22.4	*	23.4	6,901.6	25.3	*
OD-1-W1	20.3	*	21.2	19,214.2	25.1	*
OD-1-W2	28.2	5,470.4	27.3	13,559.3	30.4	*
OD-2-A	400.0	-	-	-	-	-
OD-2-C	54,867.2	-	-	-	-	-
OD-2-H	14.9	-	-	-	-	-
OD-2-W1	40,761.3	-	-	-	-	-
OD-2-W2	98.3	-	-	-	-	-

Table C.8 Median Ranks for Open Discovery on all Cases

Approach	Min		Avg		Max	
	Sum	**Max**	**Sum**	**Max**	**Sum**	**Max**
NPMI	1,211.9	1,675.4	1,173.9	1,692.8	1,156.5	1,657.4
SCP	1,342.8	1,616.5	1,291.7	1,585.4	1,293.1	1,558.8
χ^2	1,376.1	1,623.0	1,305.0	1,591.1	1,304.2	1,564.3
***t*-test**	1,172.1	1,423.1	1,163.8	1,320.1	1,149.9	1,301.9
LLR	1,205.8	1,496.1	1,137.8	1,358.1	1,132.9	1,326.4
Jaccard	1,197.3	1,547.1	1,178.5	1,477.0	1,169.9	1,431.5
Count	1,175.4	1,659.0	1,146.0	1,335.6	1,146.0	1,341.6
OD-1-A	1,911.5	1,912.0	1,909.5	1,909.5	1,908.5	1,911.7
OD-1-C	1,910.5	1,909.6	1,909.5	1,909.5	1,913.4	1,915.8
OD-1-H	1,914.3	1,912.8	1,909.5	1,909.5	1,907.5	1,910.6
OD-1-W1	1,910.6	1,910.3	1,909.5	1,909.5	1,908.3	1,911.6
OD-1-W2	1,910.3	1,910.5	1,909.5	1,909.5	1,908.3	1,914.0
OD-2-A	1,154.1	-	-	-	-	-
OD-2-C	**1,113.1**	-	-	-	-	-
OD-2-H	1,315.8	-	-	-	-	-
OD-2-W1	1,670.4	-	-	-	-	-
OD-2-W2	1,869.5	-	-	-	-	-

Table C.9 Mean Ranks (MR) for time-sliced BioGRID

Approach	Min		Avg		Max	
	Sum	**Max**	**Sum**	**Max**	**Sum**	**Max**
NPMI	2.75	1.40	2.62	1.47	2.35	0.96
SCP	2.29	1.48	1.80	1.36	1.51	0.98
χ^2	2.25	1.47	1.79	1.36	1.50	0.98
***t*-test**	2.51	1.87	2.52	1.33	2.37	1.10
LLR	2.83	1.79	2.19	1.27	1.90	1.10
Jaccard	2.86	1.44	2.57	1.43	2.12	1.04
Count	2.00	1.04	1.91	1.07	1.70	0.94
OD-1-A	1.22	1.08	1.27	1.27	1.26	1.25
OD-1-C	1.25	1.14	1.27	1.27	1.24	1.20
OD-1-H	1.24	1.12	1.27	1.27	1.25	1.24
OD-1-W1	1.21	1.11	1.27	1.27	1.26	1.25
OD-1-W2	1.21	1.11	1.27	1.27	1.26	1.25
OD-2-A	5.17	-	-	-	-	-
OD-2-C	**5.46**	-	-	-	-	-
OD-2-H	4.11	-	-	-	-	-
OD-2-W1	2.58	-	-	-	-	-
OD-2-W2	2.46	-	-	-	-	-

Table C.10 Mean MAP for time-sliced BioGRID

Approach	Min		Avg		Max	
	Sum	**Max**	**Sum**	**Max**	**Sum**	**Max**
NPMI	2.08	1.14	1.96	1.2	1.81	0.82
SCP	1.7	1.21	1.35	1.08	1.23	0.83
χ^2	1.68	1.21	1.34	1.08	1.23	0.83
***t*-test**	1.88	1.61	1.9	0.96	1.82	0.82
LLR	2.17	1.63	1.56	0.92	1.34	0.81
Jaccard	2.19	1.19	1.96	1.15	1.66	0.86
Count	1.77	1.04	1.49	0.9	1.3	0.78
OD-1-A	0.92	0.82	0.92	0.92	0.93	0.93
OD-1-C	0.92	0.85	0.92	0.92	0.94	0.93
OD-1-H	0.93	0.86	0.92	0.92	0.92	0.92
OD-1-W1	0.90	0.85	0.92	0.92	0.92	0.91
OD-1-W2	0.91	0.84	0.92	0.92	0.92	0.91
OD-2-A	3.36	-	-	-	-	-
OD-2-C	**3.42**	-	-	-	-	-
OD-2-H	2.78	-	-	-	-	-
OD-2-W1	1.75	-	-	-	-	-
OD-2-W2	1.76	-	-	-	-	-

Table C.11 Mean Mean Reciprocal Rank (MRR) for time-sliced BioGRID

Approach	Min		Avg		Max	
	Sum	**Max**	**Sum**	**Max**	**Sum**	**Max**
NPMI	2.35	1.0	2.06	1.03	1.89	0.56
SCP	1.95	1.08	1.24	0.86	1.01	0.58
χ^2	1.89	1.08	1.23	0.86	1.0	0.58
***t*-test**	1.9	1.4	1.94	0.86	1.8	0.74
LLR	2.47	1.52	1.75	0.68	1.38	0.58
Jaccard	2.47	1.05	2.18	0.89	1.64	0.54
Count	1.9	0.79	1.44	0.73	1.35	0.62
OD-1-A	0.96	0.83	0.98	0.98	1.00	0.99
OD-1-C	0.97	0.88	0.98	0.98	1.01	1.01
OD-1-H	1.00	0.88	0.98	0.98	0.96	0.96
OD-1-W1	0.95	0.89	0.98	0.98	0.97	0.97
OD-1-W2	0.97	0.86	0.98	0.98	0.98	0.97
OD-2-A	4.45	-	-	-	-	-
OD-2-C	**4.73**	-	-	-	-	-
OD-2-H	3.78	-	-	-	-	-
OD-2-W1	2.15	-	-	-	-	-
OD-2-W2	1.87	-	-	-	-	-

Table C.12 Mean Relevance-precision (R-precision) for time-sliced BioGRID