CONSTRUCTING AND MINING STRUCTURED HETEROGENEOUS INFORMATION NETWORKS FROM MASSIVE TEXT CORPORA

BY

JINGBO SHANG

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science in the Graduate College of the University of Illinois at Urbana-Champaign, 2019

Urbana, Illinois

Doctoral Committee:

Professor Jiawei Han, Chair and Director of Research
Professor Tarek Abdelzaher
Professor Jian Peng
Dr. Flip Korn, Google Research
ABSTRACT

In today’s information society, we are soaked with overwhelming amounts of natural-language text data, ranging from news articles and social media posts to research literature, medical records, and corporate reports. A grand challenge for data miners is to develop effective and scalable methods to mine such massive unstructured text corpora to discover hidden structures and generate structured heterogeneous information networks, from which actionable knowledge can be generated based on user’s need. There are three major questions as follows. Can machines automatically “digest” a given (domain-specific) corpus and identify real-world entities and their relations mentioned in the corpus? Can human experts efficiently understand and consume the sophisticated, gigantic structured networks constructed by machines? Can such machine-extracted information benefit downstream applications in various fields?

The massive and messy nature of text data poses significant challenges to creating techniques for automatic processing and algorithmic analysis of contents that scale with text volume. State-of-the-art information extraction approaches rely on heavy task-specific annotations (e.g., annotating terrorist attack-related entities in web forum posts written in Arabic) to build (deep) machine learning models. In contrast, our research harnesses “the power of massive data” and develops a family of data-driven approaches for automatic knowledge discovery. Our methods, to alleviate the need for heavy human annotation, utilize distant supervision from existing, open knowledge bases and statistical signals (e.g., frequency and point-wise mutual information) based on massive corpora. Such approaches are therefore general, extensible to texts corpora in multiple languages and across multiple domains. The goal of our research is to create general data-driven methods to transform text data of various kinds into structured databases of human knowledge.

This thesis outlines an automated framework, AutoNet, which focuses on automatically extracting structured networks of entities and relations embedded in a large-scale text corpus. In addition, it constructs a high-quality topic taxonomy for more efficient human explorations. The key philosophy of “automatic” here is to extract high-quality structured knowledge and insights with little human effort. Specifically, it first identifies corpus-wide high-quality phrases. From high-quality phrases, we distinguish typed entities and relational phrases, and further connect entities by relational phrases. In this way, we organize the entities and relations as heterogeneous information networks. Such networks extracted from massive text corpora are typically of gigantic sizes — millions of nodes and billions of edges...
would be common scenarios. Therefore, after the networks are constructed, we propose to construct a topic taxonomy to make human explorations more efficient. The topic taxonomy organizes the network in a structured way, so human experts can have a bird-eye view of the whole network and easily drill down to the particular sub-network of interests.

We attempt to make the whole AutoNet framework automated (i.e., saves human annotation effort), robust (i.e., is effective across multiple languages and domains), and scalable (i.e., works for web-scale input). In this thesis, we mainly cover automated, robust, and scalable models and real-world applications in three main problems,

- **Mining High-Quality Phrases.** We first show how to unify multiple statistical signals to estimate the phrase quality using a classifier based on weak supervision. And then, we improve the accuracy of quality estimation by rectifying the frequencies based on phrasal segmentation results. We further demonstrate that some public knowledge bases (e.g., Wikipedia) can replace the weak supervision and even lead to better results. Such phrase mining methods are purely data-driven, thus being domain-agnostic and language-independent.

- **Recognizing Named Entities.** Recent advances in deep neural models for named entity recognition have freed human effort from handcrafting features. Moving one step further, we show that using existing entity dictionaries (i.e., entity type, entity name, and some synonyms) can achieve competitive entity recognition performance as state-of-the-art supervised methods. We believe such distantly supervised entity recognition models can serve as initial deployments in various applications, and provide a solid foundation for active learning and further human annotations. It could save tremendous human effort.

- **Building Topic Taxonomies.** Different from existing methods using text data or (extracted) network structures separately, we propose to let the text collaborate with network structures. Specifically, we combine these two types of data as text-rich networks, and then construct a topic taxonomy to obtain a holistic view of all data. We employ motif patterns (i.e., subgraph patterns at the type schema level) to represent information from networks, and further conduct an instance-level selection to choose relevant information. Based on textual contexts and selected motif instances, we learn term embedding jointly from text and network, and then obtain term clusters as taxonomy nodes. Therefore, the constructed taxonomy is more accurate than those built based on text/network only.
To my family for their love and support.
ACKNOWLEDGMENTS

First of all, I would like to acknowledge my Ph.D. advisor, Jiawei Han, who sets a high standard and serves as a shining example for an excellent researcher and the best advisor. It is one of my most wise decisions ever to join Jiawei’s data mining group. It is your “real data, real problem, real solution” research principle that shapes my research philosophy. You always encourage us to lead ambitious research projects that can change the world. I still clearly remember the “CS Net” idea that you demonstrated to me at our first meeting. This “CS Net” idea motivated me to work on AutoNet (i.e., this dissertation), which constructs and mines structured knowledge and insights from massive unstructured text data. You are so generous to share most of your time with us for discussions. It is hard for outsiders to believe that Jiawei is, in fact, the most reachable person in our group, but our group members all know, it is true. You always say that we can have discussions at any time, even in the evenings or over weekends. After I start to work with some students at University of California at San Diego (UCSD), I realize how difficult it is. I really appreciate the tremendous effort that you spent for all the data mining group members. You are extremely supportive and open-minded, always encouraging me to explore new problems and emerging models, and at the same time, providing all resources needed. During my last year at University of Illinois at Urbana-Champaign (UIUC), you provided huge help on my job search, by promoting me to relevant folks at different places, providing timely feedback on my application materials and job talk rehearsals, and giving me tips for interviews. Your mentorship has certainly prepared me well for an academic position after completing my Ph.D., not only in being able to conduct productive research, but also in being an effective lecturer and advisor. I owe so much to Jiawei, and the only way to redeem myself is to keep spreading your spirits to younger generations.

I would like to thank all the other thesis committee members, Tarek Abdelzaher, Jian Peng, and Flip Korn. I collaborated with Jian in my first a few years of Ph.D. study, and he always provides me sharp comments and useful suggestions to improve our research projects. I attended Tarek’s class in 2016 and I have learned a lot of interesting knowledge about Internet of Things (IoT), which inspired many potential application scenarios of my research. I spent the summer of 2017 with Flip working on Fake News Detection at Google Research NYC — our preliminary model achieved the 4th place in an open challenge and its final version was published in CIKM’18. They have provided valuable feedback and asked insightful questions during my thesis proposal and defense. Moreover, all of them
have provided enormous help in my job search process and valuable advice for crafting a successful academic career.

I also want to express my deep gratitude to my advisors and mentors along the way: Professor Yong Yu, Dr. Tianqi Chen, Dr. Hang Li, Dr. Zhengdong Lu, and Dr. Yu Zheng. They opened the gate to machine learning and data mining research for me, when I was an undergraduate student at Shanghai Jiao Tong University. Prof. Yong Yu planted the seed of being a computer scientist in me. Tianqi taught me much about research methodology and implementation details. Hang, Zhengdong, and Yu demonstrated me role models as researchers and polished my writing substantially. They have prepared me so well for starting my Ph.D. study at UIUC.

The Ph.D. study is never only about research but also about life. I’m fortunate to have constant support from both my family and friends. My parents are the most wonderful parents, Zhiming Shang and Lijun Gao — they are always there for me. Special thanks to my wife Xin Zhao — an angel who always sheds lights into my heart when skies are gray. Without her, I cannot imagine that I could overcome all the challenges along the way and make it this far. I have also met many close friends and productive collaborators at UIUC, who share laughs with me in our “cornfields”. Alphabetically, they are: Yucheng Chen, Xiangyu Chen, Xiusi Chen, Xiaotao Gu, Jiaxin Huang, Meng Jiang, Yu Meng, Huan Gui, Qi Li, Zoey Li, De Liao, Jialu Liu, Liyuan Liu, Meng Qu, Xiang Ren, Fangbo Tao, Jiaming Shen, Yu Shi, Uttam Thakore, Wenzhu Tong, Chi Wang, Qi Wang, Xuan Wang, Frank Xu, Dandong Yin, Haizi Yu, Hongkun Yu, Quan Yuan, Chao Zhang, Xinyang Zhang, Yu Zhang, Qi Zhu, Wanzheng Zhu.
# TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION ................................. 1
   1.1 Overview and Contributions ......................... 2
   1.2 Demo System and Open-Source Tools .................. 8
   1.3 Applications in Various Domains .................... 9
   1.4 Awards and Overall Impact .......................... 10

CHAPTER 2 AUTOMATED PHRASE MINING .................. 12
   2.1 Overview and Motivations ........................... 12
   2.2 Related Work ....................................... 15
   2.3 Preliminaries ....................................... 16
   2.4 Robust Positive-Only Distant Training .............. 19
   2.5 POS-Guided Phrasal Segmentation .................... 21
   2.6 Experiments ....................................... 26
   2.7 Single-Word Phrase Extension ....................... 36
   2.8 Summary ........................................... 38

CHAPTER 3 AUTOMATED NAMED ENTITY RECOGNITION .... 40
   3.1 Overview and Motivations ........................... 40
   3.2 Related Work ....................................... 41
   3.3 AutoNER Framework .................................. 42
   3.4 Neural Models ...................................... 43
   3.5 Distant Supervision Refinement ..................... 47
   3.6 Experiments ....................................... 49
   3.7 Summary ........................................... 54

CHAPTER 4 AUTOMATED TOPIC TAXONOMY CONSTRUCTION . 55
   4.1 Overview and Motivations ........................... 55
   4.2 Related Work ....................................... 58
   4.3 Preliminary ......................................... 60
   4.4 Our Framework ...................................... 62
   4.5 Anchor Term Selection ................................ 63
   4.6 Joint Embedding from Textual and Motif Contexts . 65
   4.7 Term and Document Allocation ....................... 69
   4.8 Experiments ....................................... 69
   4.9 Summary ........................................... 78
The major goal of data mining is to extract structured knowledge and insights from big data. This goal has been pursued by generations of data mining researchers. As we all know, we are now in the big data era. Every second, we are generating tons of data, in the forms of videos, audios, images, texts, etc., In fact, the majority of data in the real world takes the form of unstructured or loosely structured text (e.g., news, scientific papers, medical records, and social media posts). And more importantly, the volume of these text data keeps growing rapidly. Can machines automatically “digest” a given (domain-specific) corpus and identify real-world entities and their relations mentioned in the corpus? Can human experts efficiently understand and consume the sophisticated, gigantic structured networks constructed by machines? Can such machine-extracted information benefit downstream
applications in various fields? To answer these questions, in this thesis, we study the problem of turning unstructured text data into structured knowledge in an automatic manner. The key philosophy of “automatic” here is to extract high-quality structured knowledge and insights with little human effort.

**Challenges.** The massive and messy nature of text data poses significant challenges to creating techniques for automatic processing and algorithmic analysis of content that scale with text volume. State-of-the-art information extraction (IE) approaches rely on heavy task-specific annotations (e.g., annotating terrorist attack-related entities in web forum posts written in Arabic) to build (deep) machine learning models. When human effort is too expensive for a complicated task, these factors become bottlenecks in the development of both supervised and rule-based methods. Recent advances in bootstrapping pattern learning (e.g., NELL [1], KnowItAll [2], OpenIE [3]) aim to reduce the amount of human involvement – only an initial set of annotated examples/patterns is required from domain experts, to iteratively produce more patterns and examples for the task. Such a process, however, requires manual intermediate spot-checking on a regular basis to avoid error propagation, and suffers from low coverage on “implicit relations”, i.e., those that are not overtly expressed in the corpus and so fail to match textual patterns generated by the systems.

In contrast, our research harnesses “the power of massive data” and develops a family of data-driven approaches for automatic knowledge discovery. My methods, to alleviate the need for heavy human annotation, utilize distant supervision from existing, open knowledge bases and statistical signals (e.g., frequency and point-wise mutual information) based on massive corpora. Such approaches are therefore general, extensible to texts corpora in multiple languages and across multiple domains. The goal of our research is to create general data-driven methods to transform text data of various kinds into structured databases of human knowledge.

1.1 OVERVIEW AND CONTRIBUTIONS

This thesis outlines an automated framework, **AutoNet**, which focuses on automatically extracting structured networks of entities and relations embedded in a large-scale text corpus. In addition, it constructs a high-quality topic taxonomy for more efficient human explorations. The key philosophy of “automatic” here is to extract high-quality structured knowledge and insights with little human effort. Specifically, it first identifies corpus-wide high-quality phrases. From high-quality phrases, we distinguish typed entities and relational phrases, and further connect entities by relational phrases. In this way, we organize the entities and relations as heterogeneous information networks. Such networks extracted from
massive text corpora are typically of gigantic sizes — millions of nodes and billions of edges would be common scenarios. Therefore, after the networks are constructed, we propose to construct a topic taxonomy to make human explorations more efficient. The topic taxonomy organizes the network in a structured way, so human experts can have a bird-eye view of the whole network and easily drill down to the particular sub-network of interests.

Let’s walk through the full pipeline using an example as follows. Suppose the text input contains millions of computer science papers and the public knowledge base is Wikipedia. We can expect to find computer science terminologies as high-quality phrases, such as “data
“mining”, “machine learning”, and “support vector machine”. During the second step, we will be able to identify different methods (e.g., “logistic regression” and “support vector machine”), problems (e.g., “binary classification”), and datasets. The relation extraction module will let us know that both “logistic regression” and “support vector machine” methods “can solve” the problem “binary classification”. Note that “can solve” is a relational phrase here. The topic taxonomy will have top-level nodes representing different research areas, such as “machine learning”, “data mining”, “database”, “computer vision”, and “natural language processing”. In the second level, there will be some sub-areas. For example, under “data mining”, there will be “recommender system”, “social network”, and “frequent pattern mining”. Such taxonomy can help the user easily navigate to the concept that he/she is interested in. It also provides the mapping to get back to the text-rich network and identify the relevant subgraph, thus making the browsing and understanding much easier.

In summary, AutoNet is a data-driven framework, which takes advantage of distant supervision from existing, open knowledge bases and statistical signals (e.g., frequency and point-wise mutual information) based on massive corpora, and thus requires no additional human curation or annotation. This framework is expected to be domain-agnostic and language-independent.

Our AutoNet framework can be applied in many domain-specific applications quickly, because it only requires public knowledge bases, without much additional expert effort. Figure 1.5 presents four examples, as follows. Given all biomedical papers from the PubMed database, with the help of MeSH term knowledge base, we can build a network of diseases,

![Figure 1.5: Applications of AutoNet in Various Domains.](image-url)
drugs, genes, and proteins, which is useful in many healthcare applications. Using news articles and Wikipedia as input, we can construct a network of relationships between different people, locations, organizations, and events, which could be helpful to social science research. If we apply AutoNet to Yelp review articles, using the Menu dictionary as the distant supervision, we can recognize dish names and their relations, which can provide useful signals for a better recommendation. If we apply AutoNet to Yahoo Finance articles, using Investopedia, which is the Wikipedia in the finance domain, the extracted structure information can provide useful signals for automated trading models.

We have contributed a series of innovative methods into this AutoNet framework, as follows.

1.1.1 Part I: Phrase Mining

As one of the fundamental tasks in text analysis, phrase mining aims at extracting quality phrases from a text corpus and has various downstream applications including information extraction/retrieval, taxonomy construction, and topic modeling. Most existing methods rely on complex, trained linguistic analyzers, and thus likely have an unsatisfactory performance on text corpora of new domains and genres without extra but expensive adaption. None of the state-of-the-art models, even data-driven models, is fully automated because they require human experts for designing rules or labeling phrases.

We developed SegPhrase [4] and AutoPhrase [5] algorithms to mine high-quality phrases, including candidate entity names and relational phrases, using weak or distant supervision. Both methods leverage corpus-level statistical signals (e.g., frequency and point-wise mutual information). AutoPhrase further utilizes the entity names from external knowledge bases (KBs) as positive examples, and employs random forest-like ensemble learning technique to estimate phrase quality scores robustly. AutoPhrase supports any language as long as a general knowledge base (e.g., Wikipedia) in that language is available, while benefiting from, but not requiring, a POS tagger. Both methods demonstrate better domain independence compared with existing methods, and generalize well on text corpora written in different languages (e.g., English, Chinese, Spanish, Japanese, and Arabic).

Contributions.

- We formulate and study an important problem, automated phrase mining, and analyze its major challenges as above.
- We propose a robust positive-only distant training method for phrase quality estimation to minimize the human effort.
• We develop a novel phrasal segmentation model to leverage POS tags to achieve further improvement, when a POS tagger is available.
• We demonstrate the robustness, accuracy, and efficiency of our method and show improvements over prior methods, with results of experiments conducted on five real-world datasets in different domains (scientific papers, business reviews, and Wikipedia articles) and different languages (English, Spanish, and Chinese).
• We successfully extend AutoPhrase to model single-word phrases, which brings about 10% to 30% recall improvements on different datasets.

1.1.2 Part II: Entity Recognition and Typing

We developed a series of deep learning models [6, 7, 8, 9, 10] to recognize entities in context and assign types to them.

Following traditional supervised setting, we have developed four models LM-LSTM-CRF [6], LD-Net [7], CrossWeigh [9], and Raw-to-End [10]. All these models are deep neural models, which allow us to build reliable named entity recognition (NER) systems without handcrafting features. Specifically, LM-LSTM-CRF [6] empowers the sequence labeling framework by co-training the LSTM-CRF model with a language model, which leverages the power of nearly unlimited raw texts. LD-Net [7] further prunes the language model for sequence labeling, which can be viewed as an efficient version of “ELMo” [11] for contextualized representation learning. CrossWeigh [9] aims to recognize the potential annotation mistakes by human annotators and let the training process be aware of them. Raw-to-End [10] focuses on noisy texts, such as tweets.

While freeing human effort from feature engineering, such methods still require large amounts of human-annotated training data. Imperfect labeled data generated by distant supervision has been found useful in a variety of tasks but remains to be further explored for NER. Therefore, we further developed AutoNER [8], which goes beyond the sequence labeling framework, develops a new tie-or-break labeling scheme to better utilize the noisy distant supervision and achieves remarkable improvements over the previous state-of-the-art model SwellShark [12], even though SwellShark requires much effort from domain experts. Specifically, we propose to tailor the dictionary based on the corpus and introduce unknown-typed quality phrases to improve the label effectiveness; In order to better leverage the unknown-typed phrases, we propose a new “Tie or Break” tag scheme and a novel neural architecture to make independent predictions. Experiments on two biomedical datasets demonstrate that, without any human annotation, AutoNER achieves competitive results with state-of-the-art supervised benchmarks.
Contributions.

• We propose AutoNER, a novel neural model with the new Tie or Break scheme for the distantly supervised NER task.

• We revise the traditional NER model to the Fuzzy-LSTM-CRF model, which serves as a strong distantly supervised baseline.

• We explore to refine distant supervision for better NER performance, such as incorporating high-quality phrases to reduce false-negative labels, and conduct ablation experiments to verify the effectiveness.

• Experiments on three benchmark datasets demonstrate that AutoNER achieves the best performance when only using dictionaries with no additional human effort and is even competitive with the supervised benchmarks.

1.1.3 Part III: Relation Extraction and Attribute Discovery

Extracting entities and their relations from text is an important task for understanding massive text corpora. Open information extraction (IE) systems mine relation tuples (i.e., entity arguments and a predicate string to describe their relation) from sentences. These relation tuples are not confined to a predefined schema for the relations of interests. However, current Open IE systems focus on modeling local context information in a sentence to extract relation tuples, while ignoring the fact that global statistics in a large corpus can be collectively leveraged to identify high-quality sentence-level extractions.

Together with our group members, we further developed ReMine [13, 14] and MetaPAD [15] to construct high-quality links and attributes in HIN. Both methods are built upon our phrase mining and entity recognition methods. MetaPAD first replaces entities by their types (e.g., replaces “Donald Trump” by “$PER”) in context, and then applies our phrase mining methods on such “meta texts” to identify entity-attribute patterns (e.g., “$PER, $DIGITS-year-old”). ReMine integrates local context signals and global structural signals in a unified, distant-supervision framework. Experiments on two real-world corpora from different domains demonstrate the effectiveness, generality, and robustness of ReMine when compared to state-of-the-art open IE systems.

Contributions.

• We propose a novel open IE framework, ReMine, that can extract relation tuples with local context and global cohesiveness.

• We develop a context-dependent phrasal segmentation algorithm that can identify high-quality phrases of multiple types.
• We design a unified objective to measure both tuple quality in a local context and global cohesiveness of candidate tuples.
• Extensive experiments on three public datasets demonstrate that ReMine achieves state-of-the-art performance on both entity phrase extraction task as well as Open IE task.

1.1.4 Part IV: Topic Taxonomy Construction

The automated construction of topic taxonomies can benefit numerous applications, including web search, recommendation, and knowledge discovery. One of the major advantages of automatic taxonomy construction is the ability to capture corpus-specific information and adapt to different scenarios. To better reflect the characteristics of a corpus, we take the meta-data of documents into consideration and view the corpus as a text-rich network.

Therefore, we recently developed NetTaxo [16], which is a novel automatic topic taxonomy construction framework, which goes beyond the existing paradigm and allows text data to collaborate with network structure. Specifically, we learn term embeddings from both text and network as contexts. Network motifs are adopted to capture appropriate network contexts. We conduct an instance-level selection for motifs, which further refines term embedding according to the granularity and semantics of each taxonomy node. Clustering is then applied to obtain sub-topics under a taxonomy node. Extensive experiments on two real-world datasets demonstrate the superiority of our method over the state-of-the-art, and further verify the effectiveness and importance of instance-level motif selection.

Contributions.
• We propose a novel topic taxonomy construction framework, NetTaxo, which integrates text data and network structures effectively and systematically.
• We design an instance-level motif selection method to choose the appropriate information from network data. Moreover, it is adaptive to the granularity and semantics of each taxonomy node.
• We conduct extensive experiments on real-world datasets to demonstrate the superiority of NetTaxo over many baselines and verify the importance and effectiveness of the instance-level motif selection.

1.2 DEMO SYSTEM AND OPEN-SOURCE TOOLS

Our developed methods are not only theoretically sound, but also practically useful. Integrating above methods (e.g., AutoPhrase, AutoNER, and ReMine) together, we have de-
ployed a *prototype system* for AutoNet [17] and demonstrated it in KDD 2018. One can find AutoNet’s *demo video* at YouTube: https://www.youtube.com/watch?v=tdtBigWq_vo&feature=youtu.be.

Our software has also attracted much attention from the open-source community. As of Oct 2019, our methods for phrase mining and named entity recognition have received **over 1,500 stars** (i.e., likes) and **over 400 forks** on GitHub as of Oct 2019. Here is a list of our open-sourced tools, related to this thesis.

- **Phrase Mining**
  - AutoPhrase: https://github.com/shangjingbo1226/AutoPhrase
  - SegPhrase: https://github.com/shangjingbo1226/SegPhrase

- **Named Entity Recognition**
  - AutoNER: https://github.com/shangjingbo1226/AutoNER
  - CrossWeigh: https://github.com/ZihanWangKi/CrossWeigh
  - Raw-to-End: https://github.com/LiyuanLucasLiu/Raw-to-End

- **Relation Extraction and Attribute Discovery**
  - ReMine: https://github.com/GentleZhu/ReMine
  - MetaPAD: https://github.com/mjiang89/MetaPAD

Our topic taxonomy construction method, NetTaxo, will be open-sourced soon after its current review process.

1.3 APPLICATIONS IN VARIOUS DOMAINS

Our methods have been widely adopted in different industries and organizations. Our phrase mining technique (SegPhrase [4] and AutoPhrase [5]) has been transferred to U.S. Army Research Lab and NIH Big Data to Knowledge Center to identify (emerging) phrases from domain-specific text corpora. These two methods have been successfully deployed in a wide range of industries, including *tech companies* (e.g., Google, Facebook, Microsoft Bing, and TripAdvisor), *financial organizations* (e.g., International Monetary Fund (IMF)), and *biomedical companies* (e.g., Invitae Corporation). Our named entity recognition technique (AutoNER [8]) is using by the tech company CooTek to detect emerging entities as trigger words for their AI-assisted functions.
We worked closely with bioinformatics researchers to apply our research to the biomedical domain. For example, our LM-LSTM-CRF and AutoNER models have been successfully adapted to the biomedical domain, generalizing well on recognizing biomedical entities, such as diseases, genes, proteins, and drugs [18, 19]. AutoNER has attracted the attention from senior investigators and researchers in National Center for Biotechnology Information (NCBI).

1.4 AWARDS AND OVERALL IMPACT

Our methods have been recognized by many prestigious awards. Our phrase mining technique (SegPhrase [4] and AutoPhrase [5]) has been awarded a Grand Prize of Yelp Dataset Challenge. The thesis work on AutoNet has been awarded a Google Ph.D. Fellowship in 2017 (sole winner of under Structured Data and Data Management in North America) and a C. W. Gear Outstanding Graduate Student Award from University of Illinois.

This thesis focuses on developing automatic methods to turn unstructured text data into structured knowledge and insights. The key philosophy of “automatic” is to achieve high performance, in terms of both effectiveness and efficiency, without any additional human annotations. We harness the power of “data redundancy” in massive texts and leverage the existing knowledge base as distant supervision at the same time. Our contributions are a series of domain-agnostic, language-independent methods that work effectively in the area of text mining and information extraction. Our work has a broad impact on numerous downstream applications, including but never limited to knowledge base construction, text-based predictive tasks (e.g., sentiment analysis), text summarization, web search and indexing, recommender systems, text-rich network mining, and many other text mining tasks. In summary, our work has been used in the following settings:

• Used in the real world:
  - Our phrase mining technique (SegPhrase [4] and AutoPhrase [5]) has been transferred to U.S. Army Research Lab and NIH Big Data to Knowledge Center to identify (emerging) phrases from domain-specific text corpora.
  - Our phrase mining technique has been successfully deployed in a wide range of industries, including tech companies (e.g., Google, Facebook, Microsoft Bing, and TripAdvisor), financial organizations (e.g., International Monetary Fund (IMF)), and biomedical companies (e.g., Invitae Corporation)
  - Our named entity recognition technique (AutoNER [8]) is using by the tech company CooTek to detect emerging entities as trigger words for their AI-assisted functions.
– Our LM-LSTM-CRF and AutoNER models have been successfully adapted to the biomedical domain, generalizing well on recognizing biomedical entities, such as diseases, genes, proteins, and drugs [19, 18].

– AutoNER has attracted the attention from senior investigators and researchers in National Center for Biotechnology Information (NCBI).

– Our methods for phrase mining and named entity recognition have received over 1,500 stars (i.e., likes) and over 400 forks on GitHub as of Oct 2019.

• Taught in classes and conference tutorials:
  – Our methods on phrase mining (SegPhrase [4] and AutoPhrase [5]), named entity recognition (LM-LSTM-CRF [7] and AutoNER [8]), and discriminative pattern-based classification (DPClass [20] and DPPred [21]) are being taught in graduate courses, e.g., University of Illinois at Urbana-Champaign (CS 512).
  – Our methods are introduced as major parts of tutorials in the top conferences of data mining, databases and information systems (SIGKDD, WWW, SIGMOD, and VLDB).
  – We have published a book “Phrase Mining from Massive Text and its Applications” [22].

• Awards:
  – Our phrase mining technique (SegPhrase [4] and AutoPhrase [5]) has been awarded a Grand Prize of Yelp Dataset Challenge.
  – The thesis work on AutoNet has been awarded a Google Ph.D. Fellowship in 2017 (sole winner of under Structured Data and Data Management in North America) and a C. W. Gear Outstanding Graduate Student Award from University of Illinois.

Next, we will discuss how to **automatically** conduct phrase mining, entity recognition, and taxonomy construction in detail one by one.
CHAPTER 2: AUTOMATED PHRASE MINING

2.1 OVERVIEW AND MOTIVATIONS

Phrase mining refers to the process of automatic extraction of high-quality phrases (e.g., scientific terms and general entity names) in a given corpus (e.g., research papers and news). Representing the text with quality phrases instead of n-grams can improve computational models for applications such as information extraction/retrieval, taxonomy construction, and topic modeling [23, 24, 25].

Let’s walk through an example together to better understand the motivation of phrase mining. Suppose we are analyzing all US news articles on April 9, 2017. As shown in Figure 2.1, one can leverage phrase mining results to better understand what’s the hot topic rather than looking at ambiguous unigram words. From the phrase cloud, one can immediately realize that the news articles are talking about the well-known United Airline incident happened at the Chicago airport — security guards punched a passenger whose name is David Dao and pulled him out from the aircraft. Therefore, there are huge benefits if we upgrade all text analysis from the unigram-level to the phrase-level using phrase mining techniques.

Almost all the state-of-the-art methods, however, require human experts at certain levels. Most existing methods [26, 27, 28] rely on complex, trained linguistic analyzers (e.g., dependency parsers) to locate phrase mentions, and thus may have unsatisfactory performance on text corpora of new domains and genres without extra but expensive adaption. Our latest domain-independent method SegPhrase [4] outperforms many other approaches [29, 26, 27, 28, 30, 31, 24], but still needs domain experts to first carefully select
hundreds of varying-quality phrases from millions of candidates, and then annotate them with binary labels.

Such reliance on manual efforts by domain and linguistic experts becomes an impediment for timely analysis of massive, emerging text corpora in specific domains. An ideal automated phrase mining method is supposed to be domain-independent, with minimal human effort\(^1\) or reliance on linguistic analyzers. Bearing this in mind, we propose a novel automated phrase mining framework AutoPhrase in this chapter, going beyond SegPhrase, to further avoid additional manual labeling effort and enhance the performance, mainly using the following two new techniques.

**Robust Positive-Only Distant Training.** In fact, many high-quality phrases are freely available in general knowledge bases, and they can be easily obtained to a scale that is much larger than that produced by human experts. Domain-specific corpora usually contain some quality phrases also encoded in general knowledge bases, even when there may be no other domain-specific knowledge bases. Therefore, for distant training, we leverage the existing high-quality phrases, as available from general knowledge bases, such as Wikipedia and Freebase, to get rid of additional manual labeling effort. We independently build samples of positive labels from general knowledge bases and negative labels from the given domain corpora, and train a number of base classifiers. We then aggregate the predictions from these classifiers, whose independence helps reduce the noise from negative labels.

**POS-Guided Phrasal Segmentation.** There is a trade-off between the accuracy and domain-independence when incorporating linguistic processors in the phrase mining method.

- On the domain independence side, the accuracy might be limited without linguistic knowledge. It is difficult to support multiple languages well, if the method is completely language-blind.

- On the accuracy side, relying on complex, trained linguistic analyzers may hurt the domain-independence of the phrase mining method. For example, it is expensive to adapt dependency parsers to special domains like clinical reports.

As a compromise, we propose to incorporate a pre-trained part-of-speech (POS) tagger to further enhance the performance, when it is available for the language of the document collection. The POS-guided phrasal segmentation leverages the shallow syntactic information in POS tags to guide the phrasal segmentation model locating the boundaries of phrases more accurately.

In principle, AutoPhrase can support any language as long as a general knowledge base in

\[^1\]The phrase “minimal human effort” indicates using only existing general knowledge bases without any other human effort.
that language is available. In fact, at least 58 languages have more than 100,000 articles in Wikipedia as of Feb, 2017\(^2\). Moreover, since pre-trained part-of-speech (POS) taggers are widely available in many languages (e.g., more than 20 languages in TreeTagger [32]\(^3\)), the POS-guided phrasal segmentation can be applied in many scenarios. It is worth mentioning that for domain-specific knowledge bases (e.g., MeSH terms in the biomedical domain) and trained POS taggers, the same paradigm applies. In this chapter, without loss of generality, we only assume the availability of a general knowledge base together with a pre-trained POS tagger. As a result, **AutoPhrase** is positioned as Figure 2.2.

As demonstrated in our experiments, **AutoPhrase** not only works effectively in multiple domains like scientific papers, business reviews, and Wikipedia articles, but also supports multiple languages, such as English, Spanish, and Chinese. In addition, **AutoPhrase** can be extended to model single-word phrases.

Our main contributions are highlighted as follows:

- We study an important problem, *automated phrase mining*, and analyze its major challenges as above.
- We propose a robust positive-only distant training method for phrase quality estimation to minimize the human effort.
- We develop a novel phrasal segmentation model to leverage POS tags to achieve further improvement, when a POS tagger is available.
- We demonstrate the robustness, accuracy, and efficiency of our method and show improvements over prior methods, with results of experiments conducted on five real-world datasets in different domains (scientific papers, business reviews, and Wikipedia articles) and different languages (English, Spanish, and Chinese).
- We successfully extend **AutoPhrase** to model single-word phrases, which brings about 10\% to 30\% recall improvements on different datasets.

---

\(^2\)https://meta.wikimedia.org/wiki/List_of_Wikipedias  
\(^3\)http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/
The rest of this chapter is organized as follows. Section 2.2 positions our work relative to existing works. Section 2.3 defines basic concepts including four requirements of phrases. The details of our method are covered in Section 2.4 and Section 2.5. Extensive experiments and case studies are presented in Section 2.6. Sections 2.7 extends AutoPhrase to model the single-word phrases and explores the effectiveness. We conclude this chapter in Section 2.8.

2.2 RELATED WORK

Identifying quality phrases efficiently has become ever more central and critical for effective handling of massively increasing-size text datasets. In contrast to keyphrase extraction [33, 34, 35, 36, 37], this task goes beyond the scope of single documents and utilizes useful cross-document signals. In [38, 39, 40], interesting phrases can be queried efficiently for ad-hoc subsets of a corpus, while the phrases are based on simple frequent pattern mining methods. The natural language processing (NLP) community has conducted extensive studies typically referred to as automatic term recognition [35, 26, 27, 28, 41], for the computational task of extracting terms (such as technical phrases). This topic also attracts attention in the information retrieval (IR) community [42, 31] since selecting appropriate indexing terms is critical to the improvement of search engines where the ideal indexing units represent the main concepts in a corpus, not just literal bag-of-words.

Text indexing algorithms typically filter out stop words and restrict candidate terms to noun phrases. With pre-defined part-of-speech (POS) rules, one can identify noun phrases as term candidates in POS-tagged documents. Supervised noun phrase chunking techniques [43, 44, 45] exploit such tagged documents to automatically learn rules for identifying noun phrase boundaries. Other methods may utilize more sophisticated NLP technologies such as dependency parsing to further enhance the precision [46, 47]. With candidate terms collected, the next step is to leverage certain statistical measures derived from the corpus to estimate phrase quality. Some methods rely on other reference corpora for the calibration of “termhood” [28]. The dependency on these various kinds of linguistic analyzers, domain-dependent language rules, and expensive human labeling, makes it challenging to extend these approaches to emerging, big, and unrestricted corpora, which may include many different domains, topics, and languages.

To overcome this limitation, data-driven approaches opt instead to make use of frequency statistics in the corpus to address both candidate generation and quality estimation [30, 31, 48, 49, 24, 4]. They do not rely on complex linguistic feature generation, domain-specific rules or extensive labeling efforts. Instead, they rely on large corpora containing hundreds of thousands of documents to help deliver superior performance [4].
In [31], several indicators, including frequency and comparison to super/sub-sequences, were proposed to extract \( n \)-grams that reliably indicate frequent, concise concepts. Deane [30] proposed a heuristic metric over frequency distribution based on Zipfian ranks, to measure lexical association for phrase candidates. As a preprocessing step towards topical phrase extraction, El-Kishky et al. [24] proposed to mine significant phrases based on frequency as well as document context following a bottom-up fashion, which only considers a part of quality phrase criteria, popularity and concordance. Our previous work [4] succeeded at integrating phrase quality estimation with phrasal segmentation to further rectify the initial set of statistical features, based on local occurrence context. Unlike previous methods which are purely unsupervised, [4] required a small set of phrase labels to train its phrase quality estimator. [50] follows [4] and further refines the phrasal segmentation. It is worth mentioning that all these approaches still depend on the human effort (e.g., setting domain-sensitive thresholds). Therefore, extending them to work automatically is challenging.

2.3 PRELIMINARIES

The goal of this chapter is to develop an automated phrase mining method to extract quality phrases from a large collection of documents without human labeling effort, and with only limited, shallow linguistic analysis. The main input to the automated phrase mining task is a corpus and a knowledge base. The input corpus is a textual word sequence in a particular language and a specific domain, of arbitrary length. The output is a ranked list of phrases with decreasing quality.

**Problem Formulation 2.1: Phrase Mining.** Given a large document corpus \( C \), which can be any textual word sequences with arbitrary lengths, such as articles, titles and queries, phrase mining tries to assign a value between 0 and 1 to indicate the quality of each phrase mentioned in \( D \) and discovers a set of quality phrases \( K = \{K_1, \cdots, K_M\} \) with their quality scores greater than 0.5, as well as to provide a segmenter for locating quality phrase mentions in any unseen text snippet.

The AutoPhrase framework is shown in Figure 2.3. The work flow is completely different from our previous domain-independent phrase mining method requiring human effort [4], although the phrase candidates and the features used during phrase quality (re-)estimation are the same. In this Chapter, we propose a robust positive-only distant training to minimize the human effort and develop a POS-guided phrasal segmentation model to improve the model performance. In this section, we briefly introduce basic concepts and components as preliminaries.
A phrase is defined as a sequence of words that appear consecutively in the text, forming a complete semantic unit in certain contexts of the given documents [51]. Compare to the entity, the phrase is a more general concept. Indeed, many high quality phrases are entities, like person names. However, there are also other phrases such as verb phrases. The phrase quality is defined to be the probability of a word sequence being a complete semantic unit, meeting the following criteria [4]:

- **Popularity**: Quality phrases should occur with sufficient frequency in the given document collection.
- **Concordance**: The collocation of tokens in quality phrases occurs with significantly higher probability than expected due to chance.
- **Informativeness**: A phrase is informative if it is indicative of a specific topic or concept.
- **Completeness**: Long frequent phrases and their subsequences within those phrases may both satisfy the 3 criteria above. A phrase is deemed complete when it can be interpreted as a complete semantic unit in some given document context. Note that a phrase and a subphrase contained within it, may both be deemed complete, depending on the context in which they appear. For example, “relational database system”, “relational database” and “database system” can all be complete in certain context.

AutoPhrase will estimate the phrase quality based on the positive and negative pools twice, once before and once after the POS-guided phrasal segmentation. That is, the POS-guided phrasal segmentation requires an initial set of phrase quality scores; we estimate the scores based on raw frequencies beforehand; and then once the feature values have been rectified, we re-estimate the scores.

Only the phrases satisfying all above requirements are recognized as quality phrases.

Example 2.1: Multi-Word Quality Phrases. Examples are shown in Table 2.1. “strong
Table 2.1: Quality of Example Multi-Word Phrases.

<table>
<thead>
<tr>
<th>Phrase</th>
<th>Quality?</th>
<th>Failure Criteria</th>
</tr>
</thead>
<tbody>
<tr>
<td>strong tea</td>
<td>✓</td>
<td>N/A</td>
</tr>
<tr>
<td>heavy tea</td>
<td>x</td>
<td>concordance</td>
</tr>
<tr>
<td>this paper</td>
<td>x</td>
<td>informative</td>
</tr>
<tr>
<td>NP-complete in the strong</td>
<td>✓</td>
<td>N/A</td>
</tr>
<tr>
<td>NP-complete in the strong</td>
<td>x</td>
<td>completeness</td>
</tr>
</tbody>
</table>

"tea" is a quality phrase while "heavy tea" fails to be due to concordance. "this paper" is a popular and concordant phrase, but is not informative in research publication corpus. "NP-complete in the strong sense" is a quality phrase while "NP-complete in the strong" fails to be due to completeness.

To automatically mine these quality phrases, the first phase of AutoPhrase (see leftmost box in Figure 2.3) establishes the set of phrase candidates that contains all n-grams over the minimum support threshold $\tau$ (e.g., 30) in the corpus. Here, this threshold refers to raw frequency of the n-grams calculated by string matching. In practice, one can also set a phrase length threshold (e.g., $n \leq 6$) to restrict the number of words in any phrase. Given a phrase candidate $w_1w_2...w_n$, its phrase quality is:

$$Q(w_1w_2...w_n) = p(\lceil w_1w_2...w_n \rceil | w_1w_2...w_n) \in [0, 1]$$

(2.1)

where $\lceil w_1w_2...w_n \rceil$ refers to the event that these words constitute a phrase. $Q(\cdot)$, also known as the phrase quality estimator, is initially learned from data based on statistical features\(^4\), such as point-wise mutual information, point-wise KL divergence, and inverse document frequency, designed to model concordance and informativeness mentioned above. Note the phrase quality estimator is computed independent of POS tags. For unigrams, we simply set their phrase quality as 1. We expect that a good quality estimator will return $Q(\text{this paper}) \approx 0$ and $Q(\text{relational database system}) \approx 1$.

Then, to address the completeness criterion, the phrasal segmentation finds the best segmentation for each sentence.

**Example 2.2: Ideal Phrasal Segmentation Results.** We present ideal phrasal segmentation results of some sentences as below. The ‘/’ characters show the places to be segmented.

\(^4\)See https://github.com/shangjingbo1226/AutoPhrase for further details.
During the **phrase quality re-estimation**, related statistical features will be re-computed based on the **rectified frequency** of phrases, which means the number of times that a phrase becomes a complete semantic unit in the identified segmentation. After incorporating the rectified frequency, the phrase quality estimator \( Q(\cdot) \) also models the **completeness** in addition to **concordance** and **informativeness**.

### Table 2.2: Raw Frequency and Rectified Frequency based on Phrasal Segmentation Results in Example 2.2.

<table>
<thead>
<tr>
<th>Phrase</th>
<th>Raw Frequency</th>
<th>Rectified Frequency</th>
</tr>
</thead>
<tbody>
<tr>
<td>great firewall</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>firewall software</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>classifier SVM</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

**Example 2.3: Rectified Frequency.** Continuing the previous example, as shown in Table 2.2, the **raw frequency** of the phrase “great firewall” is 2, but its **rectified frequency** is 1. Both the **raw frequency** and the **rectified frequency** of the phrase “firewall software” are 1. The **raw frequency** of the phrase “classifier SVM” is 1, but its **rectified frequency** is 0.

### 2.4 ROBUST POSITIVE-ONLY DISTANT TRAINING

To estimate the phrase quality score for each phrase candidate, our previous work [4] required domain experts to first carefully select hundreds of varying-quality phrases from millions of candidates, and then annotate them with binary labels. For example, for computer science papers, our domain experts provided hundreds of positive labels (e.g., “spanning tree” and “computer science”) and negative labels (e.g., “paper focuses” and “important form of”). However, creating such a label set is expensive, especially in specialized domains like clinical reports and business reviews, because this approach provides no clues for how to identify the phrase candidates to be labeled. In this section, we introduce a method that only utilizes existing general knowledge bases without any other human effort.
2.4.1 Label Pools

Public knowledge bases (e.g., Wikipedia) usually encode a considerable number of high-quality phrases in the titles, keywords, and internal links of pages. For example, by analyzing the internal links and synonyms\(^5\) in English Wikipedia, more than a hundred thousand high-quality phrases were discovered. As a result, we place these phrases in a **positive pool**.

Knowledge bases, however, rarely, if ever, identify phrases that fail to meet our criteria, what we call **inferior phrases**. An important observation is that the number of phrase candidates, based on \(n\)-grams (recall leftmost box of Figure 2.3), is huge and the majority of them are actually of inferior quality (e.g., “Francisco opera and”). In practice, based on our experiments, among millions of phrase candidates, usually, only about 10% are in good quality\(^6\). Therefore, phrase candidates that are derived from the given corpus but that fail to match any high-quality phrase derived from the given knowledge base, are used to populate a large but noisy **negative pool**.

2.4.2 Noise Reduction

Directly training a classifier based on the noisy label pools is not a wise choice: some phrases of high quality from the given corpus may have been missed (i.e., inaccurately binned into the negative pool) simply because they were not present in the knowledge base. Instead, we propose to utilize an ensemble classifier that averages the results of \(T\) independently trained base classifiers. As shown in Figure 2.4, for each base classifier, we randomly draw \(K\) phrase candidates with replacement from the positive pool and the negative pool respectively.

---

\(^5\)https://github.com/kno10/WikipediaEntities

\(^6\)This percentage is estimated when the used knowledge base is Wikipedia. It may vary when different knowledge bases are used.
(considering a canonical balanced classification scenario). This size-$2K$ subset of the full set of all phrase candidates is called a perturbed training set \cite{52}, because the labels of some ($\delta$ in the figure) quality phrases are switched from positive to negative. In order for the ensemble classifier to alleviate the effect of such noise, we need to use base classifiers with the lowest possible training errors. We grow an unpruned decision tree to the point of separating all phrases to meet this requirement. In fact, such decision tree will always reach 100\% training accuracy when no two positive and negative phrases share identical feature representations in the perturbed training set. In this case, its ideal error is $\frac{\delta}{2K}$, which approximately equals to the proportion of switched labels among all phrase candidates (i.e., $\frac{\delta}{2K} \approx 10\%$). Therefore, the value of $K$ is not sensitive to the accuracy of the unpruned decision tree and is fixed as 100 in our implementation. Assuming the extracted features are distinguishable between quality and inferior phrases, the empirical error evaluated on all phrase candidates, $p$, should be relatively small as well.

An interesting property of this sampling procedure is that the random selection of phrase candidates for building perturbed training sets creates classifiers that have statistically independent errors and similar erring probability \cite{52, 53}. Therefore, we naturally adopt random forest \cite{54}, which is verified, in the statistics literature, to be robust and efficient. The phrase quality score of a particular phrase is computed as the proportion of all decision trees that predict that phrase is a quality phrase. Suppose there are $T$ trees in the random forest, the ensemble error can be estimated as the probability of having more than half of the classifiers misclassifying a given phrase candidate as follows.

$$
\text{ensemble_error}(T) = \sum_{t=\lceil1+T/2\rceil}^{T} \binom{T}{t} p^t (1-p)^{T-t} \quad (2.2)
$$

From Figure 2.5, one can easily observe that the ensemble error is approaching 0 when $T$ grows. In practice, $T$ needs to be set larger due to the additional error brought by model bias. Empirical studies can be found in Figure 2.10.

2.5 POS-GUIDED PHRASAL SEGMENTATION

Phrasal segmentation addresses the challenge of measuring completeness (our fourth criterion) by locating all phrase mentions in the corpus and rectifying their frequencies obtained originally via string matching.

The corpus is processed as a length-$n$ POS-tagged word sequence $\Omega = \Omega_1\Omega_2\ldots\Omega_n$, where $\Omega_i$ refers to a pair consisting of a word and its POS tag $\langle w_i, t_i \rangle$. A POS-guided phrasal
segmentation is a partition of this sequence into m segments induced by a boundary index sequence \( B = \{ b_1, b_2, \ldots, b_{m+1} \} \) satisfying \( 1 = b_1 < b_2 < \ldots < b_{m+1} = n+1 \). The \( i \)-th segment refers to \( \Omega_{b_i} \Omega_{b_i+1} \ldots \Omega_{b_{i+1}-1} \).

Compared to the phrasal segmentation in our previous work [4], the POS-guided phrasal segmentation addresses the completeness requirement in a context-aware way, instead of equivalently penalizing phrase candidates of the same length. In addition, POS tags provide shallow, language-specific knowledge, which may help boost phrase detection accuracy, especially at syntactic constituent boundaries for that language.

Given the POS tag sequence for the full length-\( n \) corpus is \( t = t_1 t_2 \ldots t_n \), containing the tag subsequence \( t_l \ldots t_{r-1} \) (denote as \( t_{[l,r)} \) for clarity), the POS quality score for that tag subsequence is defined to be the conditional probability of its corresponding word sequence being a complete semantic unit. Formally, we have

\[
T(t_{[l,r)}) = p([w_l \ldots w_r] | t) \in [0,1] 
\]

The POS quality score \( T(\cdot) \) is designed to reward the phrases with their correctly identified POS sequences, as follows.

**Example 2.4: POS Quality Score.** Suppose the whole POS tag sequence is “NN NN NN VB DT NN”. A good POS sequence quality estimator might return \( T(\text{NN NN NN}) \approx 1 \) and \( T(\text{NN VB}) \approx 0 \), where NN refers to singular or mass noun (e.g., database), VB means verb in the base form (e.g., is), and DT is for determiner (e.g., the).
Algorithm 2.1: POS-Guided Phrasal Segmentation (PGPS)

1 Input: Corpus $\Omega = \Omega_1 \Omega_2 \ldots \Omega_n$, phrase quality $Q$, parameters $\theta_u$ and $\delta(t_x, t_y)$.
2 Output: Optimal boundary index sequence $B$.

// $h_i \equiv \max_B p(\Omega_1 \Omega_2 \ldots \Omega_{i-1}, B|Q, \theta, \delta)$
3 $h_1 \leftarrow 1$, $h_i \leftarrow 0$ ($1 < i \leq n + 1$)
4 for $i = 1$ to $n$ do
5      for $j = i + 1$ to $\min(i + \text{length threshold}, n + 1)$ do
6          // Efficiently implemented via Trie.
7          if there is no phrase starting with $w_{[i,j]}$ then
8              break
9          // In practice, log and addition are used to avoid underflow.
10         if $h_i \times p(j, \lceil w_{[i,j]} \rceil | i, t_{[i,j]}) > h_j$ then
11            $h_j \leftarrow h_i \times p(j, \lceil w_{[i,j]} \rceil | i, t_{[i,j]})$
12            $g_j \leftarrow i$
13      end if
14 end for
15 $j \leftarrow n + 1$, $m \leftarrow 0$
16 while $j > 1$ do
17      $m \leftarrow m + 1$
18      $b_m \leftarrow j$
19      $j \leftarrow g_j$
20 end while
21 return $B \leftarrow 1, b_m, b_{m-1}, \ldots, b_1$

The particular form of $T(\cdot)$ we have chosen is:

$$T(t_{[l,r]}) = \left(1 - \delta(t_{b_i-1}, t_{b_i})\right) \times \prod_{j=l+1}^{r-1} \delta(t_{j-1}, t_j)$$  \hspace{1cm} (2.4)

where, $\delta(t_x, t_y)$ is the probability that the POS tag $t_x$ is exactly precedes POS tag $t_y$ within a phrase in the given document collection. In this formula, the first term indicates the probability that there is a phrase boundary between the words indexed $r - 1$ and $r$, while the latter product indicates the probability that all POS tags within $t_{[l,r]}$ are in the same phrase. This POS quality score can naturally counter the bias to longer segments because $\forall i > 1$, exactly one of $\delta(t_{i-1}, t_i)$ and $(1 - \delta(t_{i-1}, t_i))$ is always multiplied no matter how the corpus is segmented. Note that the length penalty model in our previous work [4] is a special case when all values of $\delta(t_x, t_y)$ are the same.

Mathematically, $\delta(t_x, t_y)$ is defined as:

$$\delta(t_x, t_y) = p(\ldots w_1w_2\ldots | \Omega, \text{tag}(w_1) = t_x \land \text{tag}(w_2) = t_y)$$  \hspace{1cm} (2.5)
As it depends on how documents are segmented into phrases, $\delta(t_x, t_y)$ is initialized uniformly and will be learned during the phrasal segmentation.

Now, after we have both phrase quality $Q(\cdot)$ and POS quality $T(\cdot)$ ready, we are able to formally define the POS-guided phrasal segmentation model. The joint probability of a POS tagged sequence $\Omega$ and a boundary index sequence $B = \{b_1, b_2, \ldots, b_{m+1}\}$ is factorized as:

$$p(\Omega, B) = \prod_{i=1}^{m} p \left( b_{i+1}, \left\lceil w_{[b_i, b_{i+1}]} \right\rceil \bigg| b_i, t \right)$$

(2.6)

where $p(b_{i+1}, \left\lceil w_{[b_i, b_{i+1}]} \right\rceil | b_i, t)$ is the probability of observing a word sequence $w_{[b_i, b_{i+1}]}$ as the $i$-th quality segment given the previous boundary index $b_i$ and the whole POS tag sequence $t$.

Since the phrase segments function as a constituent in the syntax of a sentence, they usually have weak dependence on each other [51, 4]. As a result, we assume these segments in the word sequence are generated one by one for the sake of both efficiency and simplicity.

For each segment, given the POS tag sequence $t$ and the start index $b_i$, the generative process is defined as follows.

1. Generate the end index $b_{i+1}$, according to its POS quality

$$p(b_{i+1}|b_i, t) = T(t_{[b_i, b_{i+1}}))$$

(2.7)

2. Given the two ends $b_i$ and $b_{i+1}$, generate the word sequence $w_{[b_i, b_{i+1}]}$ according to a multinomial distribution over all segments of length-$(b_{i+1} - b_i)$.

$$p(w_{[b_i, b_{i+1}]}|b_i, b_{i+1}) = p(w_{[b_i, b_{i+1}]}|b_{i+1} - b_i)$$

(2.8)

3. Finally, we generate an indicator whether $w_{[b_i, b_{i+1}]}$ forms a quality segment according to its quality

$$p(\left\lceil w_{[b_i, b_{i+1}]} \right\rceil | w_{[b_i, b_{i+1}]} = Q(w_{[b_i, b_{i+1}]}))$$

(2.9)

We denote $p(w_{[b_i, b_{i+1}]}|b_{i+1} - b_i)$ as $\theta_{w_{[b_i, b_{i+1}]}}$, for convenience. Integrating the above three generative steps together, we have the following probabilistic factorization:

$$p(b_{i+1}, \left\lceil w_{[b_i, b_{i+1}]} \right\rceil | b_i, t)$$

$$= p(b_{i+1}|b_i, t)p(w_{[b_i, b_{i+1}]}|b_i, b_{i+1})p(\left\lceil w_{[b_i, b_{i+1}]} \right\rceil | w_{[b_i, b_{i+1}]}))$$

(2.10)

$$= T(t_{[b_i, b_{i+1}}))\theta_{w_{[b_i, b_{i+1}]}Q(w_{[b_i, b_{i+1}]}))$$

Therefore, there are three subproblems:
Algorithm 2.2: Viterbi Training (VT)

1. **Input:** Corpus $\Omega$ and phrase quality $Q$.
2. **Output:** $\theta_u$ and $\delta(t_x, t_y)$.
3. Initialize $\theta$ with normalized raw frequencies.

4. While $\theta_u$ does not converge do
   5. While $\delta(t_x, t_y)$ does not converge do
      6. $B \leftarrow$ best segmentation via Alg. 2.1
      7. Update $\delta(t_x, t_y)$ using $B$ according to Eq. (2.12)
      8. $B \leftarrow$ best segmentation via Alg. 2.1
      9. Update $\theta_u$ using $B$ according to Eq. (2.13)

10. **Return** $\theta_u$ and $\delta(t_x, t_y)$

1. Learn $\theta_u$ for each word and phrase candidate $u$;
2. Learn $\delta(t_x, t_y)$ for every POS tag pair; and
3. Infer $B$ when $\theta_u$ and $\delta(t_x, t_y)$ are fixed.

We employ the maximum a posterior principle and maximize the joint log likelihood:

$$\log p(\Omega, B) = \sum_{i=1}^{m} \log p\left( b_{i+1}, \left[ w_{[b_i, b_i+1]} \right]| b_i, t \right)$$

(2.11)

Given $\theta_u$ and $\delta(t_x, t_y)$, to find the best segmentation that maximizes Equation (2.11), we develop an efficient dynamic programming algorithm for the POS-guided phrasal segmentation as shown in Algorithm 2.1.

When the segmentation $S$ and the parameter $\theta$ are fixed, the closed-form solution of $\delta(t_x, t_y)$ is:

$$\delta(t_x, t_y) = \frac{\sum_{i=1}^{m} \sum_{j=b_i}^{b_{i+1}-2} \mathbb{1}(t_j = t_x \land t_{j+1} = t_y)}{\sum_{i=1}^{n-1} \mathbb{1}(t_i = t_x \land t_{i+1} = t_y)}$$

(2.12)

where $\mathbb{1}(\cdot)$ denotes the identity indicator. $\delta(t_x, t_y)$ is the unsegmented ratio among all $(t_x, t_y)$ pairs in the given corpus.

Similarly, once the segmentation $S$ and the parameter $\delta$ are fixed, the closed-form solution of $\theta_u$ can be derived as:

$$\theta_u = \frac{\sum_{i=1}^{m} \mathbb{1}(w_{[b_i, b_{i}+1]} = u)}{\sum_{i=1}^{m} \mathbb{1}(b_{i+1} - b_i = |u|)}$$

(2.13)

We can see that $\theta_u$ is the times that $u$ becomes a complete segment normalized by the number of the length-$|u|$ segments.

As shown in Algorithm 2.2, we choose Viterbi Training (or Hard EM in literature [55]) to iteratively optimize parameters, because Viterbi Training converges fast and results in
sparse and simple models for Hidden Markov Model-like tasks [55].

2.5.1 Complexity Analysis

The time complexity of the most time consuming components in our framework, such as frequent $n$-gram, feature extraction, POS-guided phrasal segmentation, are all $O(|\Omega|)$ with the assumption that the maximum number of words in a phrase is a small constant (e.g., $n \leq 6$), where $|\Omega|$ is the total number of words in the corpus. Therefore, AutoPhrase is linear to the corpus size and thus being very efficient and scalable. Meanwhile, every component can be parallelized in an almost lock-free way grouping by either phrases or sentences.

2.6 EXPERIMENTS

In this section, we will apply the proposed method to mine quality phrases from five massive text corpora across three domains (scientific papers, business reviews, and Wikipedia articles) and in three languages (English, Spanish, and Chinese). We compare the proposed method with many other methods to demonstrate its high performance. Then we explore the robustness of the proposed positive-only distant training and its performance against expert labeling. The advantage of incorporating POS tags in phrasal segmentation will also be proved. In the end, we present case studies.

2.6.1 Datasets

To validate that the proposed positive-only distant training can effectively work in different domains and the POS-guided phrasal segmentation can support multiple languages effectively, we have five large collections of text in different domains and languages, as shown in Table 2.3: Abstracts of English computer science papers from DBLP\textsuperscript{7}, English business reviews from Yelp\textsuperscript{8}, Wikipedia articles\textsuperscript{9} in English (EN), Spanish (ES), and Chinese (CN). From the existing general knowledge base Wikipedia, we extract popular mentions of entities by analyzing intra-Wiki citations within Wiki content\textsuperscript{10}. On each dataset, the intersection between the extracted popular mentions and the generated phrase candidates forms the positive pool. Therefore, the size of positive pool may vary in different datasets of the same language.

\textsuperscript{7}https://aminer.org/citation
\textsuperscript{8}https://www.yelp.com/dataset_challenge
\textsuperscript{9}https://dumps.wikimedia.org/
\textsuperscript{10}https://github.com/kno10/WikipediaEntities
Table 2.3: Five real-world massive text corpora in different domains and multiple languages. $|\Omega|$ is the total number of words. $size_p$ is the size of positive pool.

| Dataset | Domain            | Language | $|\Omega|$ | File size | $size_p$ |
|---------|-------------------|----------|-----------|-----------|----------|
| DBLP    | Scientific Paper  | English  | 91.6M     | 618MB     | 29K      |
| Yelp    | Business Review   | English  | 145.1M    | 749MB     | 22K      |
| EN      | Wikipedia Article | English  | 808.0M    | 3.94GB    | 184K     |
| ES      | Wikipedia Article | Spanish  | 791.2M    | 4.06GB    | 65K      |
| CN      | Wikipedia Article | Chinese  | 371.9M    | 1.56GB    | 29K      |

2.6.2 Compared Methods

We compare AutoPhrase with three lines of methods as follows. Every method returns a ranked list of phrases.

SegPhrase$^{11}$/WrapSegPhrase$^{12}$: In English domain-specific text corpora, our latest work SegPhrase outperformed phrase mining [24], keyphrase extraction [35, 31], and noun phrase chunking methods. WrapSegPhrase extends SegPhrase to different languages by adding an encoding preprocessing to first transform non-English corpus using English characters and punctuation as well as a decoding postprocessing to later translate them back to the original language. Both methods require domain expert labors. For each dataset, we ask domain experts to annotate a representative set of 300 quality/interior phrases.

Parser-based Phrase Extraction: Using complicated linguistic processors, such as parsers, we can extract minimum phrase units (e.g., NP) from the parsing trees as phrase candidates. Parsers of all three languages are available in Stanford NLP tools [56, 57, 58]. Two ranking heuristics are considered:

- TF-IDF ranks the extracted phrases by their term frequency and inverse document frequency in the given documents;
- TextRank: An unsupervised graph-based ranking model for keyword extraction [34].

Pre-trained Chinese Segmentation Models: Different from English and Spanish, phrasal segmentation in Chinese has been intensively studied because there is no space between Chinese words. The most effective and popular segmentation methods are:

- AnsjSeg$^{13}$ is a popular text segmentation algorithm for Chinese corpus. It ensembles statistical modeling methods of Conditional Random Fields (CRF) and Hidden Markov Models (HMMs) based on the n-gram setting;

---

11https://github.com/shangjingbo1226/SegPhrase
12https://github.com/remenberl/SegPhrase-MultiLingual
13https://github.com/NLPchina/ansj_seg
- JiebaPSeg\(^{14}\) is a Chinese text segmentation method implemented in Python. It builds a directed acyclic graph for all possible phrase combinations based on a prefix dictionary structure to achieve efficient phrase graph scanning. Then it uses dynamic programming to find the most probable combination based on the phrase frequency. For unknown phrases, an HMM-based model is used with the Viterbi algorithm.

Note that all parser-based phrase extraction and Chinese segmentation models are pre-trained based on general corpus.

To introduce a stronger baseline than SegPhrase and WrapSegPhrase, we introduce AutoSegPhrase, which is a hybrid of AutoPhrase and SegPhrase. AutoSegPhrase adopts the length penalty instead of \( \delta(t_x, t_y) \), while other components are the same as AutoPhrase. Meanwhile, the comparison between AutoPhrase and AutoSegPhrase can check the effectiveness of POS-guided phrasal segmentation. In addition, AutoSegPhrase is useful when there is no POS tagger.

2.6.3 Experimental Settings

**Implementation.** The preprocessing includes tokenizers from Lucene and Stanford NLP as well as the POS tagger from TreeTagger. The pre- and post-processing are in Java, while the core functions are all implemented in C++. Experiments were all conducted on a machine with 20 cores of Intel(R) Xeon(R) CPU E5-2680 v2 @ 2.80GHz. Our documented code package has been released and maintained in GitHub\(^{15}\).

**Default Parameters.** We set the minimum support threshold \( \sigma \) as 30. The maximum number of words in a phrase is set as 6 according to labels from domain experts. These are two parameters required by all methods. Other parameters required by compared methods were set according to the open-source tools or the original papers.

**Human Annotation.** We rely on human evaluators to judge the quality of the phrases which cannot be identified through any knowledge base. More specifically, on each dataset, we randomly sample 500 such phrases from the predicted phrases of each method in the experiments. These selected phrases are shuffled in a shared **pool** and evaluated by 3 reviewers independently. We allow reviewers to use search engines when unfamiliar phrases encountered. By the rule of majority voting, phrases in this pool received at least two positive annotations are **quality phrases**. The intra-class correlations (ICCs) are all more than 0.9 on all five datasets, which shows the agreement.

**Evaluation Metrics.** For a list of phrases, **precision** is defined as the number of true quality

---

\(^{14}\)https://github.com/fxsjy/jieba

\(^{15}\)https://github.com/shangjingbo1226/AutoPhrase
2.6.4 Overall Performance

Figure 2.6 and Figure 2.7 present the precision-recall curves of all compared methods evaluated by human annotation on five datasets. Overall, AutoPhrase performs the best, in terms of both precision and recall. Significant advantages can be observed, especially on two non-English datasets ES and CN. For example, on the ES dataset, the recall of AutoPhrase is about 20% higher than the second best method (SegPhrase) in absolute value. Meanwhile, there is a visible precision gap between AutoPhrase and the best baseline. The phrase chunking-based methods TF-IDF and TextRank work poorly, because the extraction and ranking are modeled separately and the pre-trained complex linguistic analyzers fail to extend to domain-specific corpora. TextRank usually starts with a higher precision than TF-IDF, but its recall is very low because of the sparsity of the constructed co-occurrence graph. TF-IDF achieves a reasonable recall but unsatisfactory precision. On the CN dataset, the pre-trained Chinese segmentation models, JiebaSeg and AnsjSeg, are very competitive, because they not only leverage training data for segmentations, but also exhaust the engineering work, including a huge dictionary for popular Chinese entity names and specific rules.
Figure 2.7: Overall Performance Evaluation in Different Languages: Precision-recall curves of all methods on three Wikipedia article datasets evaluated by human annotation. The advantages of AutoPhrase over SegPhrase are more significant in non-English languages, especially on the Chinese dataset. It is worth noting that on the Chinese dataset, AutoPhrase outperforms than two popular, pre-trained Chinese phrase extraction models. This firmly demonstrates the ability of AutoPhrase to cross the language barrier.

for certain types of entities. As a consequence, they can easily extract tons of well-known terms and people/location names. Outperforming such a strong baseline further confirms the effectiveness of AutoPhrase.

The comparison among the English datasets across three domains (i.e., scientific papers, business reviews, and Wikipedia articles) demonstrates that AutoPhrase is reasonably domain-independent. The performance of parser-based methods TF-IDF and TextRank depends on the rigorous degree of the documents. For example, it works well on the DBLP dataset but poorly on the Yelp dataset. However, without any human effort, AutoPhrase can work effectively on domain-specific datasets, and even outperforms SegPhrase, which is supervised by the domain experts.

The comparison among the Wikipedia article datasets in three languages (i.e., EN, ES, and CN) shows that, first of all, AutoPhrase supports multiple languages. Secondly, the advantage of AutoPhrase over SegPhrase/WrapSegPhrase is more obvious on two non-English datasets ES and CN than the EN dataset, which proves that the helpfulness of introducing the POS tagger.

As conclusions, AutoPhrase is able to support different domains and support multiple languages with minimal human effort.
Figure 2.8: AUC curves of four variants when we have enough positive labels in the positive pool EP. Overall, human annotations lead to better results because they are more clean. However, similar trends between EPEN and DPEN show that the positive pool generated from knowledge bases has reasonable quality; the similar trends between EPEN and EPDN proves that our proposed robust positive-only distant training method works well. DPDN is the worst in this case but it has a great potential to be better as the size of positive pool grows.

2.6.5 Distant Training Exploration

To compare the distant training and domain expert labeling, we experiment with the domain-specific datasets DBLP and Yelp. To be fair, all the configurations in the classifiers are the same except for the label selection process. More specifically, we come up with four training pools:

1. EP means that domain experts give the positive pool.
2. DP means that a sampled subset from existing general knowledge forms the positive pool.
3. EN means that domain experts give the negative pool.
4. DN means that all unlabeled (i.e., not in the positive pool) phrase candidates form the negative pool.

By combining any pair of the positive and negative pools, we have four variants, EPEN (in SegPhrase), DPDN (in AutoPhrase), EPDN, and DPEN.

First of all, we evaluate the performance difference in the two positive pools. Compared to EPEN, DPEN adopts a positive pool sampled from knowledge bases instead of the well-designed positive pool given by domain experts. The negative pool EN is shared. As shown in Figure 2.8, we vary the size of the positive pool and plot their AUC curves. We can find that EPEN outperforms DPEN and the trends of curves on both datasets are similar.
Therefore, we conclude that the positive pool generated from knowledge bases has reasonable quality, although its corresponding quality estimator works slightly worse.

Secondly, we verify that whether the proposed noise reduction mechanism works properly. Compared to EPEN, EPDN adopts a negative pool of all unlabeled phrase candidates instead of the well-designed negative pool given by domain experts. The positive pool $EP$ is shared. In Figure 2.8, the clear gap between them and the similar trends on both datasets show that the noisy negative pool is slightly worse than the well-designed negative pool, but it still works effectively.

As illustrated in Figure 2.8, DPDN has the worst performance when the size of positive pools are limited. However, distant training can generate much larger positive pools, which may significantly beyond the ability of domain experts considering the high expense of labeling. Consequently, we are curious whether the distant training can finally beat domain experts when positive pool sizes become large enough. We call the size at this tipping point as the \textit{ideal number}.

We increase positive pool sizes and plot AUC curves of DPEN and DPDN, while EPEN and EPDN are degenerated as dashed lines due to the limited domain expert abilities. As shown in Figure 2.9, with a large enough positive pool, distant training is able to beat expert labeling. On the DBLP dataset, the ideal number is about 700, while on the Yelp dataset, it becomes around 1600. Our guess is that the ideal training size is proportional to the number of words (e.g., 91.6M in DBLP and 145.1M in Yelp). We notice that compared
to the corpus size, the ideal number is relatively small, which implies the distant training should be effective in many domain-specific corpora as if they overlap with Wikipedia.

Besides, Figure 2.9 shows that when the positive pool size continues growing, the AUC score increases but the slope becomes smaller. The performance of distant training will be finally stable when a relatively large number of quality phrases were fed.

We are curious how many trees (i.e., $T$) is enough for DPDN. We increase $T$ and plot AUC curves of DPDN. As shown in Figure 2.10, on both datasets, as $T$ grows, the AUC scores first increase rapidly and later the speed slows down gradually, which is consistent with the theoretical analysis in Section 2.4.2.

2.6.6 POS-guided Phrasal Segmentation

We are also interested in how much performance gain we can obtain from incorporating POS tags in this segmentation model, especially for different languages. We select Wikipedia article datasets in three different languages: $EN$, $ES$, and $CN$.

Figure 2.11 compares the results of AutoPhrase and AutoSegPhrase, with the best baseline methods as references. AutoPhrase outperforms AutoSegPhrase even on the English dataset $EN$, though it has been shown the length penalty works reasonably well in English [4]. The Spanish dataset $ES$ has similar observation. Moreover, the advantage of AutoPhrase becomes more significant on the $CN$ dataset, indicating the poor generality of length penalty.

In summary, thanks to the extra context information and syntactic information for the particular language, incorporating POS tags during the phrasal segmentation can work better
Figure 2.11: Comparison Between Phrase Mining Methods with/without POS tags (AutoPhrase and AutoSegPhrase) as input. Datasets in different languages are used for the comparison. The best baseline in each dataset is provided as a reference. The results show POS-guided phrasal segmentation works more smoothly in different languages. As the original segmentation method without POS information is designed for English, it works well for English and Spanish but relatively poorly on the Chinese data.

than equally penalizing phrases of the same length.

2.6.7 Case Study

We present a case study about the extracted phrases as shown in Table 2.4. The top ranked phrases are mostly named entities, which makes sense for the Wikipedia article datasets. Even in the long tail part, there are still many high-quality phrases. For example, we have the 〈great spotted woodpecker〉 (a type of birds) and 〈计算机 科学技术〉 (i.e., Computer Science and Technology) ranked about 100,000. In fact, we have more than 345K and 116K phrases with a phrase quality higher than 0.5 on the EN and CN datasets respectively.

2.6.8 Efficiency Evaluation

To study both time and memory efficiency, we choose the three largest datasets: EN, ES, and CN.

Figures 2.12(a) and 2.12(b) evaluate the running time and the peak memory usage of AutoPhrase using 10 threads on different proportions of three datasets respectively. Both time and memory are linear to the size of text corpora. Moreover, AutoPhrase can also be parallelized in an almost lock-free way and shows a linear speedup in Figure 2.12(c).

Besides, compared to the previous state-of-the-art phrase mining method SegPhrase and its variants WrapSegPhrase on three datasets, as shown in Table 2.5, AutoPhrase achieves
Table 2.4: The results of AutoPhrase on the EN and CN datasets, with translations and explanations for Chinese phrases. The whitespaces on the CN dataset are inserted by the Chinese tokenizer. It worths a mention that the general knowledge base only provides about 29K quality phrases in the positive pool and AutoPhrase is able to discover new quality phrases even in the rank of 100K. This implies that AutoPhrase has a power to discover more than 200% new quality phrases than the provided positive pool.

<table>
<thead>
<tr>
<th>Rank</th>
<th>EN Phrase</th>
<th>CN Phrase</th>
<th>Translation (Explanation)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Elf Aquitaine</td>
<td>江苏舜天</td>
<td>(the name of a soccer team)</td>
</tr>
<tr>
<td>2</td>
<td>Arnold Sommerfeld</td>
<td>苦艾酒</td>
<td>Absinthe</td>
</tr>
<tr>
<td>3</td>
<td>Eugene Wigner</td>
<td>白发魔女</td>
<td>(the name of a novel/TV-series)</td>
</tr>
<tr>
<td>4</td>
<td>Tarpon Springs</td>
<td>笔记型电脑</td>
<td>notebook computer, laptop</td>
</tr>
<tr>
<td>5</td>
<td>Sean Astin</td>
<td>党委书记</td>
<td>Secretary of Party Committee</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>20,001</td>
<td>ECAC Hockey</td>
<td>非洲国家</td>
<td>African countries</td>
</tr>
<tr>
<td>20,002</td>
<td>Sacramento Bee</td>
<td>左翼党</td>
<td>The Left (German: Die Linke)</td>
</tr>
<tr>
<td>20,003</td>
<td>Bering Strait</td>
<td>菲沙河谷</td>
<td>Fraser Valley</td>
</tr>
<tr>
<td>20,004</td>
<td>Jacknife Lee</td>
<td>海马体</td>
<td>Hippocampus</td>
</tr>
<tr>
<td>20,005</td>
<td>WXYZ-TV</td>
<td>斋贺光希</td>
<td>Mitsuki Saiga (a voice actress)</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>99,994</td>
<td>John Gregson</td>
<td>计算机科学技术</td>
<td>Computer Science and Technology</td>
</tr>
<tr>
<td>99,995</td>
<td>white-tailed eagle</td>
<td>恒天然</td>
<td>Fonterra (a company)</td>
</tr>
<tr>
<td>99,996</td>
<td>rhombic dodecahedron</td>
<td>中国作家协会副主席</td>
<td>The Vice President of Writers Association of China</td>
</tr>
<tr>
<td>99,997</td>
<td>great spotted woodpecker</td>
<td>维他命b</td>
<td>Vitamin B</td>
</tr>
<tr>
<td>99,998</td>
<td>David Manners</td>
<td>舆论导向</td>
<td>controlled guidance of the media</td>
</tr>
<tr>
<td></td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

about 8 to 11 times speedup and about 5 to 7 times memory usage improvement. These improvements are made by a more efficient indexing and a more thorough parallelization.
Figure 2.12: Efficiency evaluation of AutoPhrase on the three largest datasets. Both the running time and the peak memory are linear to the corpus size. Because of an almost lock-free parallelized implementation, the multi-threading speedup is close to linear.

Table 2.5: Efficiency comparison between AutoPhrase and SegPhrase/WrapSegPhrase utilizing 10 threads. The difference is mainly caused by a more efficient indexing and a more thorough parallelization.

<table>
<thead>
<tr>
<th></th>
<th>EN</th>
<th></th>
<th>ES</th>
<th></th>
<th>CN</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Time (mins)</td>
<td>Memory (GB)</td>
<td>Time (mins)</td>
<td>Memory (GB)</td>
<td>Time (mins)</td>
<td>Memory (GB)</td>
</tr>
<tr>
<td>AutoPhrase</td>
<td>32.77</td>
<td>13.77</td>
<td>54.05</td>
<td>16.42</td>
<td>9.43</td>
<td>5.74</td>
</tr>
<tr>
<td>(Wrap)SegPhrase</td>
<td>369.53</td>
<td>97.72</td>
<td>452.85</td>
<td>92.47</td>
<td>108.58</td>
<td>35.38</td>
</tr>
<tr>
<td>Speedup/Saving</td>
<td>11.27</td>
<td>86%</td>
<td>8.37</td>
<td>82%</td>
<td>11.50</td>
<td>83%</td>
</tr>
</tbody>
</table>

2.7 SINGLE-WORD PHRASE EXTENSION

AutoPhrase can be extended to model single-word phrases, which can gain about 10% to 30% recall improvements on different datasets. To study the effect of modeling quality single-word phrases, we choose the three Wikipedia article datasets in different languages: EN, ES, and CN.

2.7.1 Quality Estimation

In the paper, the definition of quality phrases and the evaluation only focus on multi-word phrases. In linguistic analysis, however, a phrase is not only a group of multiple words, but also possibly a single word, as long as it functions as a constituent in the syntax of a sentence [51]. As a great portion (ranging from 10% to 30% on different datasets based on our experiments) of high-quality phrases, we should take single-word phrases (e.g., [UIUC], [Illinois], and [USA]) into consideration as well as multi-word phrases to achieve a high
Figure 2.13: Precision-recall curves evaluated by human annotation with both single-word and multi-word phrases in pools. The most significant recall gap can be observed in the Chinese dataset because the ratio of quality single-word phrases is highest in Chinese.

recall in phrase mining.

Considering the criteria of quality phrases, because single-word phrases cannot be decomposed into two or more parts, the concordance and completeness are no longer definable. Therefore, we revise the requirements for quality single-word phrases as below.

- **Popularity**: Quality phrases should occur with sufficient frequency in the given document collection.

- **Informativeness**: A phrase is informative if it is indicative of a specific topic or concept.

- **Independence**: A quality single-word phrase is more likely a complete semantic unit in the given documents.

Only single-word phrases satisfying all popularity, independence, and informativeness requirements are recognized as quality single-word phrases.

<table>
<thead>
<tr>
<th>Single-Word Phrase</th>
<th>Quality?</th>
<th>Failure Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>UIUC</td>
<td>✓</td>
<td>N/A</td>
</tr>
<tr>
<td>this</td>
<td>×</td>
<td>informativeness</td>
</tr>
<tr>
<td>united</td>
<td>×</td>
<td>independence</td>
</tr>
</tbody>
</table>

**Example 2.5: Quality Single-Word Phrases.** Examples are shown in Table 2.6. “UIUC” is a quality single-word phrase. “this” is not a quality phrase due to its low informativeness. “united”, usually occurring within other quality multi-word phrases such as “United States”, “United Kingdom”, “United Airlines”, and “United Parcel Service”, is not a quality single-word phrase, because its independence is not enough.
After the phrasal segmentation, in replacement of concordance features, the independence feature is added for single-word phrases. Formally, it is the ratio of the rectified frequency of a single-word phrase given the phrasal segmentation over its raw frequency. Quality single-word phrases are expected to have large values. For example, “united” is likely to an almost zero ratio.

We use AutoPhrase+ to denote the extended AutoPhrase with quality single-word phrase estimation.

2.7.2 Experiments

We have a similar human annotation as that in the paper. Differently, we randomly sampled 500 Wiki-uncovered phrases from the returned phrases (both single-word and multi-word phrases) of each method in experiments of the paper. Therefore, we have new pools on the EN, ES, and CN datasets. The intra-class correlations (ICCs) are all more than 0.9, which shows the agreement.

Figure 2.13 compare all methods based these new pools. Note that all methods except for SegPhrase/WrapSegPhrase extract single-word phrases as well.

Significant recall advantages can be always observed on all EN, ES, and CN datasets. The recall differences between AutoPhrase+ and AutoPhrase, ranging from 10% to 30% sheds light on the importance of modeling single-word phrases. Across two Latin language datasets, EN and ES, AutoPhrase+ and AutoPhrase overlaps in the beginning, but later, the precision of AutoPhrase drops earlier and has a lower recall due to the lack of single-word phrases. On the CN dataset, AutoPhrase+ and AutoPhrase has a clear gap even in the very beginning, which is different from the trends on the EN and ES datasets, which reflects that single-word phrases are more important in Chinese. The major reason behind is that there are a considerable number of high-quality phrases (e.g., person names) in Chinese have only one token after tokenization.

2.8 SUMMARY

In this chapter, we present an automated phrase mining framework with two novel techniques: the robust positive-only distant training and the POS-guided phrasal segmentation incorporating part-of-speech (POS) tags, for the development of an automated phrase mining framework AutoPhrase. Our extensive experiments show that AutoPhrase is domain-independent, outperforms other phrase mining methods, and supports multiple languages (e.g., English, Spanish, and Chinese) effectively, with minimal human effort.
Besides, the inclusion of quality single-word phrases (e.g., [UIUC] and [USA]) which leads to about 10% to 30% increased recall and the exploration of better indexing strategies and more thorough parallelization, which leads to about 8 to 11 times running time speedup and about 80% to 86% memory usage saving over SegPhrase. Interested readers may try our released code at GitHub. We have also extended this method to more languages, such as Japanese and Arabic.

For future work, it is interesting to (1) refine quality phrases to entity mentions, (2) apply AutoPhrase to more languages, and (3) for those languages without general knowledge bases, seek an unsupervised method to generate the positive pool from the corpus, even with some noise.
CHAPTER 3: AUTOMATED NAMED ENTITY RECOGNITION

3.1 OVERVIEW AND MOTIVATIONS

Recently, extensive efforts have been made on building reliable named entity recognition (NER) models without handcrafting features [6, 59, 60]. However, most existing methods require large amounts of manually annotated sentences for training supervised models (e.g., neural sequence models) [6, 59, 60, 61]. This is particularly challenging in specific domains, where domain-expert annotation is expensive and/or slow to obtain.

To alleviate human effort, distant supervision has been applied to automatically generate labeled data, and has gained successes in various natural language processing tasks, including phrase mining [5], entity recognition [62, 12, 63], aspect term extraction [64], and relation extraction [65]. Meanwhile, open knowledge bases (or dictionaries) are becoming increasingly popular, such as WikiData and YAGO in the general domain, as well as MeSH and CTD in the biomedical domain. The existence of such dictionaries makes it possible to generate training data for NER at a large scale without additional human effort.

Problem Formulation 3.1: Automated Named Entity Recognition (AutoNER).
Given an entity dictionary and a raw corpus, we aim to learn a NER model. The entity dictionary usually has three columns: entity type, canonical name, possible synonyms. The synonyms are never required to be complete. The learned NER model can also recognize unseen entities in new corpus.

Existing distantly supervised NER models usually tackle the entity span detection problem by heuristic matching rules, such as POS tag-based regular expressions [62, 12] and exact string matching [64, 63]. In these models, every unmatched token will be tagged as non-entity. However, as most existing dictionaries have limited coverage on entities, simply ignoring unmatched tokens may introduce false-negative labels (e.g., “prostaglandin synthesis” in Fig. 3.1). Therefore, we propose to extract high-quality out-of-dictionary phrases from the corpus, and mark them as potential entities with a special “unknown” type. Moreover, every entity span in a sentence can be tagged with multiple types, since two entities of different types may share the same surface name in the dictionary. To address these challenges, we propose and compare two neural architectures with customized tagging schemes.

We start with adjusting models under the traditional sequence labeling framework. Typically, NER models are built upon conditional random fields (CRF) with the IOB or IOBES tagging scheme [6, 59, 60, 66, 61]. However, such design cannot deal with multi-label to-
kens. Therefore, we customize the conventional CRF layer in LSTM-CRF [60] into a Fuzzy CRF layer, which allows each token to have multiple labels without sacrificing computing efficiency.

To adapt to imperfect labels generated by distant supervision, we go beyond the traditional sequence labeling framework and propose a new prediction model. Specifically, instead of predicting the label of each single token, we propose to predict whether two adjacent tokens are tied in the same entity mention or not (i.e., broken). The key motivation is that, even the boundaries of an entity mention are mismatched by distant supervision, most of its inner ties are not affected, and thus more robust to noise. Therefore, we design a new Tie or Break tagging scheme to better exploit the noisy distant supervision. Accordingly, we design a novel neural architecture that first forms all possible entity spans by detecting such ties, then identifies the entity type for each span. The new scheme and neural architecture form our new model, AutoNER, which proves to work better than the Fuzzy CRF model in our experiments.

We summarize our major contributions as

- We propose AutoNER, a novel neural model with the new Tie or Break scheme for the distantly supervised NER task.
- We revise the traditional NER model to the Fuzzy-LSTM-CRF model, which serves as a strong distantly supervised baseline.
- We explore to refine distant supervision for better NER performance, such as incorporating high-quality phrases to reduce false-negative labels, and conduct ablation experiments to verify the effectiveness.
- Experiments on three benchmark datasets demonstrate that AutoNER achieves the best performance when only using dictionaries with no additional human effort and is even competitive with the supervised benchmarks.

We release all code and data for future studies\(^1\). Related open tools can serve as the NER module of various domain-specific systems in a plug-in-and-play manner.

3.2 RELATED WORK

The task of supervised named entity recognition (NER) is typically embodied as a sequence labeling problem. Conditional random fields (CRF) models built upon human annotations and handcrafted features are the standard [61, 67, 68]. Recent advances in neural models have freed domain experts from handcrafting features for NER tasks. [60, 59, 6]. Such

\(^1\)https://github.com/shangjingbo1226/AutoNER
neural models are increasingly common in the domain-specific NER tasks [69, 70, 19]. Semi-supervised methods have been explored to further improve the accuracy by either augmenting labeled datasets with word embeddings or bootstrapping techniques in tasks like gene name recognition [71, 72, 73]. Unlike these existing approaches, our study focuses on the distantly supervised setting without any expert-curated training data.

Distant supervision has attracted many attentions to alleviate human efforts. Originally, it was proposed to leverage knowledge bases to supervise relation extraction tasks [74, 65]. AutoPhrase has demonstrated powers in extracting high-quality phrases from domain-specific corpora like scientific papers and business reviews [5] but it cannot categorize phrases into typed entities in a context-aware manner. We incorporate the high-quality phrases to enrich the domain-specific dictionary.

There are attempts on the distantly supervised NER task recently [62, 12, 63, 64]. For example, SwellShark [12], specifically designed for biomedical NER, leverages a generative model to unify and model noise across different supervision sources for named entity typing. However, it leaves the named entity span detection to a heuristic combination of dictionary matching and part-of-speech tag-based regular expressions, which require extensive expert effort to cover many special cases. Other methods [62, 63] also utilize similar approaches to extract entity span candidates before entity typing. Distant-LSTM-CRF [64] has been proposed for the distantly supervised aspect term extraction, which can be viewed as an entity recognition task of a single type for business reviews. As shown in our experiments, our models can outperform Distant-LSTM-CRF significantly on the laptop review dataset.

To the best of our knowledge, AutoNER is the most effective model that can learn NER models by using, and only using dictionaries without any additional human effort.

3.3 AUTONER FRAMEWORK

Our goal, in this chapter, is to learn a named entity tagger using, and only using dictionaries. Each dictionary entry consists of 1) the surface names of the entity, including a canonical name and a list of synonyms; and 2) the entity type. Considering the limited coverage of dictionaries, we extend existing dictionaries by adding high-quality phrases as potential entities with unknown type. More details on refining distant supervision for better NER performance will be presented in Section 3.5.

Given a raw corpus and a dictionary, we first generate entity labels (including unknown labels) by exact string matching, where conflicted matches are resolved by maximizing the total number of matched tokens [75, 76, 77, 63].

Based on the result of dictionary matching, each token falls into one of three categories:
Thus, indomethacin, by inhibition of prostaglandin synthesis, may diminish.

Figure 3.1: The illustration of the Fuzzy CRF layer with modified IOBES tagging scheme. The named entity types are \{Chemical, Disease\}. “indomethacin” is a matched Chemical entity and “prostaglandin synthesis” is an unknown-typed high-quality phrase. Paths from Start to End marked as purple form all possible label sequences given the distant supervision.

1) it belongs to an entity mention with one or more known types; 2) it belongs to an entity mention with unknown type; and 3) It is marked as non-entity.

Accordingly, we design and explore two neural models, Fuzzy-LSTM-CRF with the modified IOBES scheme and AutoNER with the Tie or Break scheme, to learn named entity taggers based on such labels with unknown and multiple types. We will discuss the details in Sec. 3.4.

3.4 NEURAL MODELS

In this section, we introduce two prediction models for the distantly supervised NER task, one under the traditional sequence labeling framework and another with a new labeling scheme.

3.4.1 Fuzzy-LSTM-CRF with Modified IOBES

State-of-the-art named entity taggers follow the sequence labeling framework using IOB or IOBES scheme [66], thus requiring a conditional random field (CRF) layer to capture the dependency between labels. However, both the original scheme and the conventional CRF layer cannot handle multi-typed or unknown-typed tokens. Therefore, we propose the modified IOBES scheme and Fuzzy CRF layer accordingly, as illustrated in Figure 3.1.

Modified IOBES. We define the labels according to the three token categories. 1) For a token marked as one or more types, it is labeled with all these types and one of \{I, B, E, S\} according to its positions in the matched entity mention. 2) For a token with unknown type, all five \{I, O, B, E, S\} tags are possible. Meanwhile, all available types are assigned.
For example, when there are only two available types (e.g., Chemical and Disease), it has nine (i.e., $4 \times 2 + 1$) possible labels in total. 3) For a token that is annotated as non-entity, it is labeled as 0.

**Example 3.1: Fuzzy IOBES Labels.** As demonstrated in Fig. 3.1, based on the dictionary matching results, “indomethacin” is a singleton Chemical entity and “prostaglandin synthesis” is an unknown-typed high-quality phrase. Therefore, “indomethacin” is labeled as S-Chemical, while both “prostaglandin” and “synthesis” are labeled as 0, B-Disease, I-Disease, ..., and S-Chemical because the available entity types are \{Chemical, Disease\}. The non-entity tokens, such as “Thus” and “by”, are labeled as 0.

**Fuzzy-LSTM-CRF.** We revise the LSTM-CRF model [60] to the Fuzzy-LSTM-CRF model to support the modified IOBES labels.

Given a word sequence $(X_1, X_2, \ldots, X_n)$, it is first passed through a word-level BiLSTM [78] (i.e., forward and backward LSTMs). After concatenating the representations from both directions, the model makes independent tagging decisions for each output label. In this step, the model estimates the score $P_{i, y_j}$ for the word $X_i$ being the label $y_j$.

We follow previous works [6, 59, 60] to define the score of the predicted sequence, the score of the predicted sequence $(y_1, y_2, \ldots, y_n)$ is defined as:

$$s(X, y) = \sum_{i=0}^{n} \Phi_{y_i, y_{i+1}} + \sum_{i=1}^{n} P_{i, y_i}$$

where, $\Phi_{y_i, y_{i+1}}$ is the transition probability from a label $y_i$ to its next label $y_{i+1}$. $\Phi$ is a $(k + 2) \times (k + 2)$ matrix, where $k$ is the number of distinct labels. Two additional labels start and end are used (only used in the CRF layer) to represent the beginning and end of a sequence, respectively.

The conventional CRF layer maximizes the probability of the only valid label sequence. However, in the modified IOBES scheme, one sentence may have multiple valid label sequences, as shown in Fig. 3.1. Therefore, we extend the conventional CRF layer to a fuzzy CRF model. Instead, it maximizes the total probability of all possible label sequences by enumerating both the IOBES tags and all matched entity types. Mathematically, we define the optimization goal as Eq. 3.2.

$$p(y|X) = \frac{\sum_{\hat{y} \in Y_{\text{possible}}} e^{s(X, \hat{y})}}{\sum_{\hat{y} \in Y_X} e^{s(X, \hat{y})}}$$

where $Y_X$ means all the possible label sequences for sequence $X$, and $Y_{\text{possible}}$ contains all the
possible label sequences given the labels of modified IOBES scheme. Note that, when all labels and types are known and unique, the fuzzy CRF model is equivalent to the conventional CRF.

During the training process, we maximize the log-likelihood function of Eq. 3.2. For inference, we apply the Viterbi algorithm to maximize the score of Eq. 3.1 for each input sequence.

3.4.2 AutoNER with “Tie or Break”

Identifying the nature of the distant supervision, we go beyond the sequence labeling framework and propose a new tagging scheme, **Tie or Break**. It focuses on the ties between adjacent tokens, i.e., whether they are tied in the same entity mentions or broken into two parts. Accordingly, we design a novel neural model for this scheme.

**“Tie or Break” Tagging Scheme.** Specifically, for every two adjacent tokens, the connection between them is labeled as: (1) **Tie**, when the two tokens are matched to the same entity; (2) **Unknown**, if at least one of the tokens belongs to an unknown-typed high-quality phrase; (3) **Break**, otherwise. Tokens between every two consecutive **Break** form a token span. Each token span is associated with all its matched types, the same as for the modified IOBES scheme. For those token spans without any associated types, we assign them the additional type **None**.

**Example 3.2: Tie or Break Labels.** An example can be found in Fig. 3.2. The distant supervision shows that “ceramic unibody” is a matched **AspectTerm** and “8GB RAM” is an unknown-typed high-quality phrase. Therefore, a **Tie** is labeled between “ceramic” and “unibody”, while **Unknown** labels are put before “8GB”, between “8GB” and “RAM”, and
after “RAM”. All the other connections between two adjacent tokens are then annotated as Break.

Every two consecutive Breaks form an entity span. We will mark the type of “ceramic unibody” as AspectTerm. All other spans are annotated as None, as there is no match from the dictionary. Note that, “8GB RAM” now is a part of “and 8GB RAM ...”, therefore, marking it as None is safe.

We believe this new scheme can better exploit the knowledge from dictionary according to the following two observations. First, even though the boundaries of an entity mention are mismatched by distant supervision, most of its inner ties are not affected. More interestingly, compared to multi-word entity mentions, matched unigram entity mentions are more likely to be false-positive labels. However, such false-positive labels will not introduce incorrect labels with the Tie or Break scheme, since either the unigram is a true entity mention or a false positive, it always brings two Break labels around.

AutoNER Neural Architecture. In the Tie or Break scheme, entity spans and entity types are encoded into two folds. Therefore, we separate the entity span detection and entity type prediction into two steps.

For entity span detection, we build a binary classifier to distinguish Break from Tie, while Unknown positions will be skipped. Specifically, as shown in Fig. 3.2, for the prediction between \( i \)-th token and its previous token, we concatenate the output of the BiLSTM as a new feature vector, \( u_i \). \( u_i \) is then fed into a sigmoid layer, which estimates the probability that there is a Break as

\[
p(y_i = \text{Break}|u_i) = \sigma(w^T u_i) \tag{3.3}
\]

where \( y_i \) is the label between the \( i \)-th and its previous tokens, \( \sigma \) is the sigmoid function, and \( w \) is the sigmoid layer’s parameter. The entity span detection loss is then computed as follows.

\[
\mathcal{L}_{\text{span}} = \sum_{i|y_i \neq \text{Unknown}} l(y_i, p(y_i = \text{Break}|u_i)) \tag{3.4}
\]

Here, \( l(\cdot, \cdot) \) is the logistic loss. Note that those Unknown positions are skipped.

After obtaining candidate entity spans, we further identify their entity types, including the None type for non-entity spans. As shown in Fig. 3.2, the output of the BiLSTM will be re-aligned to form a new feature vector, which is referred as \( v_i \) for \( i \)-th span candidate. \( v_i \) will be further fed into a softmax layer, which estimates the entity type distribution as

\[
p(t_j|v_i) = \frac{e^{t_j^T v_i}}{\sum_{t_k \in L} e^{t_k^T v_i}} \tag{3.5}
\]
where $t_j$ is an entity type and $L$ is the set of all entity types including None.

Since one span can be labeled as multiple types, we mark the possible set of types for $i$-th entity span candidate as $L_i$. Accordingly, we modify the cross-entropy loss as follows.

$$
\mathcal{L}_{\text{type}} = H(\hat{p}(\cdot|v_i, L_i), p(\cdot|v_i))
$$

Here, $H(p, q)$ is the cross entropy between $p$ and $q$, and $\hat{p}(t_j|v_i, L_i)$ is the soft supervision distribution. Specifically, it is defined as:

$$
\hat{p}(t_j|v_i, L_i) = \frac{\delta(t_j \in L_i) \cdot e^{t_j^Tv_i}}{\sum_{t_k \in L} \delta(t_k \in L_i) \cdot e^{t_k^Tv_i}}
$$

where $\delta(t_j \in L_i)$ is the boolean function of checking whether the $i$-th span candidate is labeled as the type $t_j$ in the distant supervision.

It’s worth mentioning that AutoNER has no CRF layer and Viterbi decoding, thus being more efficient than Fuzzy-LSTM-CRF for inference.

3.4.3 Remarks on “Unknown” Entities

“Unknown” entity mentions are not the entities of other types, but the tokens that we are less confident about their boundaries and/or cannot identify their types based on the distant supervision. For example, in Figure 1, “prostaglandin synthesis” is an “unknown” token span. The distant supervision cannot decide whether it is a Chemical, a Disease, an entity of other types, two separate single-token entities, or (partially) not an entity. Therefore, in the FuzzyCRF model, we assign all possible labels for these tokens.

In our AutoNER model, these “unknown” positions have undefined boundary and type losses, because (1) they make the boundary labels unclear; and (2) they have no type labels. Therefore, they are skipped.

3.5 DISTANT SUPERVISION REFINEMENT

In this section, we present two techniques to refine the distant supervision for better named entity taggers. Ablation experiments in Sec. 3.6.4 verify their effectiveness empirically.
3.5.1 Corpus-Aware Dictionary Tailoring

In dictionary matching, blindly using the full dictionary may introduce false-positive labels, as there exist many entities beyond the scope of the given corpus but their aliases can be matched. For example, when the dictionary has a non-related character name “Wednesday Addams”\(^2\) and its alias “Wednesday”, many Wednesday’s will be wrongly marked as persons. In an ideal case, the dictionary should cover, and only cover entities occurring in the given corpus to ensure a high precision while retaining a reasonable coverage.

As an approximation, we tailor the original dictionary to a corpus-related subset by excluding entities whose canonical names never appear in the given corpus. The intuition behind is that to avoid ambiguities, people will likely mention the canonical name of the entity at least once. For example, in the biomedical domain, this is true for 88.12\%, 95.07\% of entity mentions on the BC5CDR and NCBI datasets respectively. We expect the NER model trained on such tailored dictionary will have a higher precision and a reasonable recall compared to that trained on the original dictionary.

3.5.2 Unknown-Typed High-Quality Phrases

Another issue of the distant supervision is about the false-negative labels. When a token span cannot be matched to any entity surface names in the dictionary, because of the limited coverage of dictionaries, it is still difficult to claim it as non-entity (i.e., negative labels) for sure. Specifically, some high-quality phrases out of the dictionary may also be potential entities.

We utilize the state-of-the-art distantly supervised phrase mining method, AutoPhrase \([5]\), with the corpus and dictionary in the given domain as input. AutoPhrase only requires unlabeled text and a dictionary of high-quality phrases. We obtain quality multi-word and single-word phrases by posing thresholds (e.g., 0.5 and 0.9 respectively). In practice, one can find more unlabeled texts from the same domain (e.g., PubMed papers and Amazon laptop reviews) and use the same domain-specific dictionary for the NER task. In our experiments, for the biomedical domain, we use the titles and abstracts of 686,568 PubMed papers (about 4\%) uniformly sampled from the whole PubTator database as the training corpus. For the laptop review domain, we use the Amazon laptop review dataset\(^3\), which is designed for the aspect-based sentiment analysis \([79]\).

We treat out-of-dictionary phrases as potential entities with “unknown” type and incorporate them as new dictionary entries. After this, only token spans that cannot be matched

\(^2\)https://en.wikipedia.org/wiki/Wednesday_Addams
\(^3\)http://times.cs.uic.edu/~wang296/Data/
Table 3.1: Dataset Overview.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>BC5CDR</th>
<th>NCBI-Disease</th>
<th>LaptopReview</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain</td>
<td>Biomedical</td>
<td>Biomedical</td>
<td>Technical Review</td>
</tr>
<tr>
<td>Entity Types</td>
<td>Disease, Chemical</td>
<td>Disease</td>
<td>AspectTerm</td>
</tr>
<tr>
<td>Dictionary</td>
<td>MeSH + CTD</td>
<td>MeSH + CTD</td>
<td>Computer Terms</td>
</tr>
<tr>
<td>Raw Sent. #</td>
<td>20,217</td>
<td>7,286</td>
<td>3,845</td>
</tr>
</tbody>
</table>

in this extended dictionary will be labeled as non-entity. Being aware of these high-quality phrases, we expect the trained NER tagger should be more accurate.

3.6 EXPERIMENTS

We conduct experiments on three benchmark datasets to evaluate and compare our proposed Fuzzy-LSTM-CRF and AutoNER with many other methods. We further investigate the effectiveness of our proposed refinements for the distant supervision and the impact of the number of distantly supervised sentences.

3.6.1 Experimental Settings

Datasets are briefly summarized in Table 3.1. More details as as follows.

- **BC5CDR** is from the most recent BioCreative V Chemical and Disease Mention Recognition task. It has 1,500 articles containing 15,935 Chemical and 12,852 Disease mentions.
- **NCBI-Disease** focuses on Disease Name Recognition. It contains 793 abstracts and 6,881 Disease mentions.
- **LaptopReview** is from the SemEval 2014 Challenge, Task 4 Subtask 1 [80] focusing on laptop aspect term (e.g., “disk drive”) Recognition. It consists of 3,845 review sentences and 3,012 AspectTerm mentions.

All datasets are publicly available. The first two datasets are already partitioned into three subsets: a training set, a development set, and a testing set. For the LaptopReview dataset, we follow [64] and randomly select 20% from the training set as the development set. Only raw texts are provided as the input of distantly supervised models, while the gold training set is used for supervised models.
Domain-Specific Dictionary. For the biomedical datasets, the dictionary is a combination of both the MeSH database\(^4\) and the CTD Chemical and Disease vocabularies\(^5\). The dictionary contains 322,882 Chemical and Disease entity surfaces. For the laptop review dataset, the dictionary has 13,457 computer terms crawled from a public website\(^6\).

Metric. We use the micro-averaged $F_1$ score as the evaluation metric. Meanwhile, precision and recall are presented. The reported scores are the mean across five different runs.

Parameters and Model Training. Based on the analysis conducted in the development set, we conduct optimization with the stochastic gradient descent with momentum. We set the batch size and the momentum to 10 and 0.9. The learning rate is initially set to 0.05 and will be shrunk by 40\% if there is no better development $F_1$ in the recent 5 rounds. Dropout of a ratio 0.5 is applied in our model. For a better stability, we use gradient clipping of 5.0. Furthermore, we employ the early stopping in the development set.

Pre-trained Word Embeddings. For the biomedical datasets, we use the pre-trained 200-dimension word vectors\(^7\) from [81], which are trained on the whole PubMed abstracts, all the full-text articles from PubMed Central (PMC), and English Wikipedia. For the laptop review dataset, we use the GloVe 100-dimension pre-trained word vectors\(^8\) instead, which are trained on the Wikipedia and GigaWord.

3.6.2 Compared Methods

Dictionary Match is our proposed distant supervision generation method. Specifically, we apply it to the testing set directly to obtain entity mentions with exactly the same surface name as in the dictionary. The type is assigned through a majority voting. By comparing with it, we can check the improvements of neural models over the distant supervision itself.

SwellShark, in the biomedical domain, is arguably the best distantly supervised model, especially on the BC5CDR and NCBI-Disease datasets [12]. It needs no human annotated data, however, it requires extra expert effort for entity span detection on building POS tagger, designing effective regular expressions, and hand-tuning for special cases.

Distant-LSTM-CRF achieved the best performance on the LaptopReview dataset without annotated training data using a distantly supervised LSTM-CRF model [64].

\(^4\)https://www.nlm.nih.gov/mesh/download_mesh.html
\(^5\)http://ctdbase.org/downloads/
\(^6\)https://www.computerhope.com/jargon.htm
\(^7\)http://bio.nlplab.org/
\(^8\)https://nlp.stanford.edu/projects/glove/
### Table 3.2: [Biomedical Domain] NER Performance Comparison

The supervised benchmarks on the BC5CDR and NCBI-Disease datasets are LM-LSTM-CRF and LSTM-CRF respectively [19]. SwellShark has no annotated data, but for entity span extraction, it requires pre-trained POS taggers and extra human efforts of designing POS tag-based regular expressions and/or hand-tuning for special cases.

<table>
<thead>
<tr>
<th>Method</th>
<th>Human Effort</th>
<th>BC5CDR</th>
<th>NCBI-Disease</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Pre</td>
<td>Rec</td>
</tr>
<tr>
<td>Supervised Benchmark</td>
<td>Gold Annotations</td>
<td>88.84</td>
<td>85.16</td>
</tr>
<tr>
<td>SwellShark</td>
<td>Regex Design + Special Case Tuning</td>
<td>86.11</td>
<td>82.39</td>
</tr>
<tr>
<td></td>
<td>Regex Design</td>
<td>84.98</td>
<td>83.49</td>
</tr>
<tr>
<td>Dictionary Match</td>
<td>None</td>
<td>93.93</td>
<td>58.35</td>
</tr>
<tr>
<td>Fuzzy-LSTM-CRF</td>
<td>None</td>
<td>88.27</td>
<td>76.75</td>
</tr>
<tr>
<td>AutoNER</td>
<td></td>
<td>88.96</td>
<td>81.00</td>
</tr>
</tbody>
</table>

### Table 3.3: [Technical Review Domain] NER Performance Comparison

The supervised benchmark refers to the challenge winner.

<table>
<thead>
<tr>
<th>Method</th>
<th>LaptopReview</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pre</td>
</tr>
<tr>
<td>Supervised Benchmark</td>
<td>84.80</td>
</tr>
<tr>
<td>Distant-LSTM-CRF</td>
<td>74.03</td>
</tr>
<tr>
<td>Dictionary Match</td>
<td>90.68</td>
</tr>
<tr>
<td>Fuzzy-LSTM-CRF</td>
<td>85.08</td>
</tr>
<tr>
<td>AutoNER</td>
<td>72.27</td>
</tr>
</tbody>
</table>

**Supervised benchmarks** on each dataset are listed to check whether AutoNER can deliver competitive performance. On the BC5CDR and NCBI-Disease datasets, LM-LSTM-CRF [6] and LSTM-CRF [60] achieve the state-of-the-art $F_1$ scores without external resources, respectively [19]. On the LaptopReview dataset, we present the scores of the Winner in the SemEval2014 Challenge Task 4 Subtask 1 [80].

### 3.6.3 NER Performance Comparison

We present $F_1$, precision, and recall scores on all datasets in Table 3.2 and Table 3.3. From both tables, one can find the AutoNER achieves the best performance when there is no extra human effort. Fuzzy-LSTM-CRF does have some improvements over the Dictionary
Table 3.4: Ablation Experiments for Dictionary Refinement. The dictionary for the LaptopReview dataset contains no alias, so the corpus-aware dictionary tailoring is not applicable.

<table>
<thead>
<tr>
<th>Method</th>
<th>BC5CDR</th>
<th></th>
<th></th>
<th>NCBI-Disease</th>
<th></th>
<th></th>
<th>LaptopReview</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pre</td>
<td>Rec</td>
<td>F1</td>
<td>Pre</td>
<td>Rec</td>
<td>F1</td>
<td>Pre</td>
<td>Rec</td>
<td>F1</td>
</tr>
<tr>
<td>AutoNER w/ Original Dict</td>
<td>82.79</td>
<td>70.40</td>
<td>76.09</td>
<td>53.14</td>
<td>63.54</td>
<td>57.87</td>
<td>69.96</td>
<td>49.85</td>
<td>58.21</td>
</tr>
<tr>
<td>AutoNER w/ Tailored Dict</td>
<td>84.57</td>
<td>70.22</td>
<td>76.73</td>
<td>77.30</td>
<td>58.54</td>
<td>66.63</td>
<td>Not Applicable</td>
<td></td>
<td></td>
</tr>
<tr>
<td>AutoNER w/ Tailored Dict &amp; Phrases</td>
<td>88.96</td>
<td>81.00</td>
<td>84.8</td>
<td>79.42</td>
<td>71.98</td>
<td>75.52</td>
<td>72.27</td>
<td>59.79</td>
<td>65.44</td>
</tr>
</tbody>
</table>

Match, but it is always worse than AutoNER.

Even though SwellShark is designed for the biomedical domain and utilizes much more expert effort, AutoNER outperforms it in almost all cases. The only outlier happens on the NCBI-disease dataset when the entity span matcher in SwellShark is carefully tuned by experts for many special cases.

It is worth mentioning that AutoNER beats Distant-LSTM-CRF, which is the previous state-of-the-art distantly supervised model on the LaptopReview dataset.

Moreover, AutoNER’s performance is competitive to the supervised benchmarks. For example, on the BC5CDR dataset, its F1 score is only 2.16% away from the supervised benchmark.

3.6.4 Distant Supervision Explorations

We investigate the effectiveness of the two techniques that we proposed in Sec. 3.5 via ablation experiments. As shown in Table 3.4, using the tailored dictionary always achieves better F1 scores than using the original dictionary. By using the tailored dictionary, the precision of the AutoNER model will be higher, while the recall will be retained similarly. For example, on the NCBI-Disease dataset, it significantly boosts the precision from 53.14% to 77.30% with an acceptable recall loss from 63.54% to 58.54%. Moreover, incorporating unknown-typed high-quality phrases in the dictionary enhances every score of AutoNER models significantly, especially the recall. These results match our expectations well.

3.6.5 Test F1 Scores vs. Size of Raw Corpus

Furthermore, we explore the change of test F1 scores when we have different sizes of distantly supervised texts. We sample sentences uniformly random from the given raw corpus and then evaluate AutoNER models trained on the selected sentences. We also study what
will happen when the gold training set is available. The curves can be found in Figure 3.3. The X-axis is the number of distantly supervised training sentences while the Y-axis is the $F_1$ score on the testing set.

When using distant supervision only, one can observe a significant growing trend of test $F_1$ score in the beginning, but later the increasing rate slows down when there are more and more raw texts.

When the gold training set is available, the distant supervision is still helpful to AutoNER. In the beginning, AutoNER works worse than the supervised benchmarks. Later, with enough distantly supervised sentences, AutoNER outperforms the supervised benchmarks. We think there are two possible reasons: (1) The distant supervision puts emphasis on those matchable entity mentions; and (2) The gold annotation may miss some good but matchable entity mentions. These may guide the training of AutoNER to a more generalized model, and thus have a higher test $F_1$ score.

### 3.6.6 Comparison with Gold Supervision

To demonstrate the effectiveness of distant supervision, we try to compare our method with gold annotations provided by human experts.

Specifically, we conduct experiments on the BC5CDR dataset by sampling different amounts of annotated articles for model training. As shown in Figure 3.4, we found that our method outperforms the supervised method by a large margin when less training examples are available. For example, when there are only 50 annotated articles available, the test F1 score drops substantially to 74.29%. To achieve a similar test F1 score (e.g., 83.91%) as our AutoNER model’s (i.e., 84.8%), the supervised benchmark model requires at least 300 annotated articles. Such results indicate the effectiveness and usefulness of AutoNER on the
scenario without sufficient human annotations.

Still, we observe that, when the supervised benchmark is trained with all annotations, it achieves the performance better than AutoNER. We conjugate that this is because AutoNER lacks more advanced techniques to handle distant supervision, and we leave further improvements of AutoNER to the future work.

3.7 SUMMARY

In this chapter, we explore how to learn an effective NER model by using, and only using dictionaries. We design two neural architectures, Fuzzy-LSTM-CRF model with a modified IOBES tagging scheme and AutoNER with a new Tie or Break scheme. In experiments on three benchmark datasets, AutoNER achieves the best $F_1$ scores without additional human efforts. Its performance is even competitive to the supervised benchmarks with full human annotation. In addition, we discuss how to refine the distant supervision for better NER performance, including incorporating high-quality phrases mined from the corpus as well as tailoring dictionary according to the given corpus, and demonstrate their effectiveness in ablation experiments.

In future, we plan to further investigate the power and potentials of the AutoNER model with Tie or Break scheme in different languages and domains. Also, the proposed framework can be further extended to other sequence labeling tasks, such as noun phrase chunking. Moreover, going beyond the classical NER setting in this paper, it is interesting to further explore distant supervised methods for the nested and multiple typed entity recognitions in the future.
4.1 OVERVIEW AND MOTIVATIONS

Constructing high-quality topic taxonomies for document collections is an important task. A topic taxonomy is a tree-structured hierarchy, where each taxonomy node contains a set of semantically similar terms. A high-quality topic taxonomy benefits various downstream applications, such as search and indexing [82], personalized content recommendation [83], and question answering [84]. For example, organizing copious scientific papers into a well-structured taxonomy gives researchers a bird’s-eye view of the field, and then they can quickly identify their interests, and easily acquire desired information [82]. A high-quality taxonomy for business reviews on Yelp can facilitate more accurate recommendations and improve user’s browsing experience.

Different applications usually require different taxonomies, therefore, automatic taxonomy construction capturing corpus-specific information becomes beneficial. The last decade has witnessed an explosive growth of digital document collections. By linking documents with their meta-data, we can view any document collection as a text-rich network. As illustrated in Figure 4.1, a collection of scientific papers can be viewed as a text-rich network with interconnected venue, author, term and paper nodes, and raw texts are associated with the paper nodes. Similarly, reviews from online platforms like Yelp and TripAdvisor can be seen as a part of a text-rich network with nodes of businesses, users, and reviews.

While most existing methods solely rely on text data [85, 86, 87, 88], incorporating network structures can bring additional, valuable information to text. Let’s use the computer science paper collection to convey our intuition. The term “frequent pattern” appears along with
“transaction database” frequently. Judging only from text data, one may put this term into the database community. However, information embedded in the network structure, such as its associated venues (e.g., “SIGKDD”) and authors (e.g., “Charu C. Aggarwal”), indicates the strong relatedness between the term “frequent pattern” and the data mining community, enabling us to assign it to the right taxonomy node.

![Diagram of motif patterns](image)

**Figure 4.2: Example Motif Patterns and Motif Instances.** (a) This motif pattern suggests that two terms are similar when they are from the papers published by the same author pairs. (b) The two terms are connected by a motif instance of the motif patterns in (a), which has two authors instantiated. The shades indicate two full instantiations of the motif pattern. (c) The other motif patterns that we used in the DBLP-5 dataset, including meta-path shaped patterns.

Acknowledging that network provides useful information for taxonomy construction, how to effectively integrate network and text remains a major challenge. We leverage motif patterns in our framework to extract useful features from the heterogeneous text-rich network. Meta-paths [89] and motif patterns [90, 91] have been widely adopted to extract useful structural information from network. As illustrated in Example 4.2, motifs are subgraph patterns that capture higher-order connectivity and precise semantics. We observe two issues of applying motif patterns in our problem. First, motif patterns are not created equal. Some motif patterns are more useful in identifying top-level concepts, while other motif patterns are better at differentiating finer concepts. Second, even only looking at one motif pattern, its motif instances are by no means equally informative. Some of them could even interfere the taxonomy construction, leading to a worse result. For example, using the motif pattern in Example 4.2(a) which captures co-authorship, some of its instances may be occasional and coincidental collaborations, thus will not help much when constructing the scientific taxonomy. To address these two issues, we propose a novel instance-level motif selection mechanism, which is specifically tailored to current node’s granularity and semantics. We show in our experiments that such selection mechanism is crucial especially when the network is relatively noisy.

We propose NetTaxo, a hierarchical embedding and clustering framework for automatic
topic taxonomy construction. The general workflow is sketched in Figure 4.3. To begin with, we ask the user to provide a set of motif patterns as guidance. This set is never assumed to be clean and equally effective. At each taxonomy node, we propose to learn term embedding from both text and network data, and then apply a soft clustering method to obtain term clusters. We first obtain initial term clusters based on term embedding learned on text data. An inter-cluster comparative analysis is then conducted to select the most representative terms as anchor terms from each cluster. We make an assumption that a helpful motif instance should have the ability to separate one cluster’s anchor terms from others. Building upon this assumption, we further distill the motif instances to include those that are relevant to the clustering, thus avoiding to introduce noise from network data. After that, we combine textual context and selected motif instances to learn term embedding jointly. Final clusters are then decided based on such joint embedding.

Experimental results demonstrate the success of our instance-level motif selection. For example, we show that, for a collection of computer science papers, at the top level of the taxonomy construction, our method locates the venue of publication (e.g., “SIGKDD”) as a strong indicator of research fields (e.g., “data mining”). Drilling down to lower levels of the taxonomy, our objective becomes to distinguish research sub-areas. Our proposed method identified specific author groups as more useful signals, such as “Cheng-Wei Wu” and “Philip S. Yu” — All their collaborations focus on the topic of high-utility itemset discovery.

To our best knowledge, this is the first work that bridges text and network data for automatic construction of topic taxonomy. Our contributions can be summarized as follows.

- We propose a novel topic taxonomy construction framework, NetTaxo, which integrates text data and network structures effectively and systematically.
• We design an instance-level motif selection method to choose the appropriate information from network data. Moreover, it is adaptive to the granularity and semantics of each taxonomy node.

• We conduct extensive experiments on real-world datasets to demonstrate the superiority of NetTaxo over many baselines and verify the importance and effectiveness of the instance-level motif selection.

4.2 RELATED WORK

We group related work into different types as follows.

4.2.1 Hyponymy-based Methods

Taxonomies have been designed to group entities into hierarchies where each node is a concept term and each parent-child pair expresses a hyponymy (a.k.a. “is-a”) relation (e.g., panda “is-a” mammal). In order to construct such taxonomies automatically, researchers have developed a number of pattern-based methods. Typically, these methods first acquire hyponymy relations from text data using lexical patterns (e.g., “A such as B”), and then organize the extracted pairs into a taxonomy by applying algorithms like maximum spanning tree. The lexical patterns are either manually designed for specific corpus [92, 93, 94, 95] or derived from corpus using some supervision or seeds [96, 97, 98, 99, 15, 100]. Such patterns have demonstrated their effectiveness at finding hyponymy relations, however, they are not suitable for constructing a topic taxonomy as (1) each node in a topic taxonomy is a cluster of terms instead of a single concept term, and (2) pattern-based methods often suffer from low recall due to the large variation of expressions in natural language on hyponymy relations.

Recently, term embedding has been widely adopted in automatic topic taxonomy construction. A common practice is to first learn term embedding from text data and then organize them into a structure based on their representation similarity [101] and cluster separation measures [102]. Utilizing pairwise hyponymy relation labels, taxonomic relations between terms and clusters can be identified through supervised models, for example, semantic projection in the embedding space [85] and neural network classifier [86]. In our setting, there are no hyponymy labels.
4.2.2 Term Clustering-based Methods

A number of clustering methods have been proposed towards automatic topic taxonomy construction from text corpora. In pioneer studies, hierarchical topic modeling [103, 104, 105, 106, 107] and bottom-up agglomerative clustering-based [108] methods are arguably the most popular and effective frameworks, before word embedding techniques become mature.

Among unsupervised frameworks using term embedding, top-down hierarchical clustering methods [88, 87] achieve the state-of-the-art. For example, TaxoGen [88] learns local term embedding from the documents associated with a taxonomy node, and then clusters terms at a deeper level. Most of these methods, including TaxoGen, only utilize the information embedded in text data, but ignore the underlying network structures in digital document collections. In our NetTaxo framework, we follow the top-down, local embedding approach but go beyond and leverage network structures to significantly improve the quality of clustering.

4.2.3 Network Clustering-based Methods

CATHYHIN [106] is arguably the state-of-the-art method solely based on network structures to construct topic taxonomies automatically. Specifically, with unigram words as a part of its node set, it attempts to mine terms (i.e., phrases) and clusters simultaneously. It ignores the context of the words, thus sacrifice the abundant information embedded in the text data, yielding unsatisfactory results in our experiments.

Another related thread is clustering algorithms on heterogeneous information networks (i.e., networks of typed nodes and edges) [109, 110]. For example, NetClus [110] starts with user-provided seed nodes and applies authority ranking together with node clustering to cluster nodes. We adopt a similar authority ranking process as a part of our motif instance selection.

4.2.4 Network Motifs

Network motifs are higher-order subgraph structures that are critical in complex networks across various domains, such as neuroscience [111], bioinformatics [90], and information networks [91]. In the context of heterogeneous information networks, network motifs, sometimes also referred to as meta-graphs, can offer more flexibility and capture richer network semantics than the widely used meta-path [89] patterns. Recent studies have shown that incorporating motifs for node embedding leads to superior performance [112, 113, 114] compared to conventional path-based methods [115, 116]. In this chapter, the quality of term
embedding is the key to the overall quality of the constructed taxonomy. While taking advantage of network motifs in our embedding learning, we further select a subset of motif instances according to the current taxonomy node. This novel approach enables us to refine the rich semantics captured by network motifs, generating embeddings better suitable for taxonomy construction.

4.3 PRELIMINARY

In this section, we first introduce the preliminary concepts and then formulate the problem by specifying the input and output.

4.3.1 Topic Taxonomy

**Topic taxonomy** is a tree-structured hierarchy $\mathcal{H}$, where each node $c \in \mathcal{H}$ contains a small set of terms $\mathcal{T}_c \subset \mathcal{T}$, which are semantically coherent and represent a conceptual topic. Moreover, the parent-child nodes in $\mathcal{H}$ should follow the topic-subtopic relation. That is, suppose a node $c$ has a set of children $\mathcal{S}_c = c_1, c_2, \ldots, c_n$, then each $c_i (1 \leq i \leq n)$ should be a sub-topic of $c$ and of the same granularity as its siblings in $\mathcal{S}_c$.

Note that, one term may belong to multiple conceptual topics and thus appear in multiple nodes. For example, “deep learning” could be a part of both “deep learning theory in machine learning” and “deep learning models in computer vision”; “data stream” could belong to “stream data indexing in database” and “stream data classification in data mining”.

4.3.2 Text-Rich Network

As mentioned before, a document collection with meta-data can be naturally viewed as a **text-rich network**, consisting of text data and network structure:

- **Text Data**: A corpus $\mathcal{D}$ and a set of terms $\mathcal{T}$. $\mathcal{T}$ includes terms in $\mathcal{D}$, which can be either specified by users or extracted from the corpus. In our experiments, we form the term set $\mathcal{T}$ by extracting high-quality phrases from the corpus $\mathcal{D}$ using AutoPhrase [5].

- **Network Structure**: A heterogeneous information network $G = (V, E, \phi, \psi)$, where $V$ is the node set and $E$ is the edge set. Type mapping $\phi$ and $\psi$ map each node $v$ to its type $\phi(v)$ and each edge $e$ to a relation $\psi(e)$. 
4.3.3 Motif Patterns

A **motif pattern** \(\Omega\) refers to a subgraph pattern at the meta level (i.e., every node is abstracted by its type). In this chapter, we study only the motif patterns having at least one node of *term* type. A **motif instance** \(m\) is an instantiation of a motif pattern by replacing the node types with concrete values. Example 4.2 presents some examples. We define “open” nodes as those single-degree nodes except for the term node, playing a role of connecting two terms. We say that two terms are connected following a motif pattern, if and only if both terms appear in motif instances sharing the same values at those “open” nodes. Therefore, we represent motif instances only by the values of “open” nodes. As an example, in Example 4.2(b), the motif instances linking to the terms “social network” and “information cascade” are the same. Both motif instances can be represented by the combination of two authors (i.e., “Jure Leskovec” and “Jon Kleinberg”).

It is worth noting that meta-path [89] can be viewed as a special case of motif patterns when they degenerate to lines. For example, the meta-path describing the shared venue relation between two terms is equivalent to the 2nd motif pattern in Example 4.2(c). The only “open” node in this motif pattern is the venue node.

(a) An example motif pattern. (b) Two terms connected by a motif instance. (c) Some other motif patterns. Meta-paths are special cases of motif patterns.

**Figure 4.4:** User-Provided Motif Patterns in Our Experiments — The DBLP-5 Dataset.

(b) Two terms connected by a motif instance.

(b) Two terms connected by a motif instance.

**Figure 4.5:** User-Provided Motif Patterns in Our Experiments — The Yelp-5 Dataset.
Example 4.1: User-provided motif patterns in our experiments. Here, we present all user-provided motif patterns in our experiments. All motif patterns used in the DBLP-5 dataset as illustrated before in Figure 4.2. For the ease of reading, we present it here again in Figure 4.4. All motif patterns used in the Yelp-5 dataset are visualized in Figure 4.5. The most complex pattern indicates a term mentioned by two users under the same business.

In this chapter, we explore to construct a topic taxonomy with a text-rich network as input. In addition, we ask user to provide a set of motif patterns as the guidance to incorporate information from network. However, the use-provided set can be noisy and we will conduct a motif instance-level selection later. Therefore, our problem is:

Problem Formulation 4.1: Automated Topic Taxonomy Construction. Given a text-rich network and a set of user-provided motif patterns as the guidance, our goal is to construct a tree-structured taxonomy hierarchy $H$, i.e., a topic taxonomy. There is no assumption that the user-provide motif patterns are all good and effective. We will conduct selections at the motif instance level.

4.4 OUR FRAMEWORK

NetTaxo is a top-down, recursive framework. Our main goal is to allocate terms into sub-topics at each taxonomy node. The allocation module relies on term embedding that is jointly learned from textual and motif contexts. We use local embedding and motif instance selection to refine the textural and motif contexts respectively.

To support our local embedding and motif instance selection module, we associate every taxonomy node with a set of weighted documents. Specifically, we maintain a weight $w_{c,d} \in [0, 1]$ for each document $d$ at the taxonomy node $c$. The weights are initialized to 1 for all documents in the root node. Alongside with term allocation, we also allocate documents from a taxonomy node to its children nodes. During the allocation process, we update $w_{c,d}$ for documents in the children nodes $c_1, c_2, \ldots, c_n$.

Figure 4.3 gives an overview of NetTaxo. At each taxonomy node, the system needs to determine the sub-topics, and then distribute terms and documents into its children accordingly. The key contribution of NetTaxo is our designed effective way of leveraging both text data and network structures.

Based on our observations and previous work [88], using term embedding learned from textual contexts alone can cluster sub-topics roughly, although not necessarily perfect. Therefore, we decide to leverage such clustering results as initialization to our subsequent motif
instance selection step. Specifically, we first follow previous work [88] to learn local term embedding and obtain initial term clusters. To be more accurate, we conduct a comparative analysis between clusters to select the most representative terms from each cluster to serve as anchor terms. Such anchor terms can be viewed as consolidated clustering information. Based on the anchor terms, we choose the appropriate motif instances. After that, we learn the term embedding jointly from the text data and the selected motif instances, which will in turn yield better clustering results. Before recursing to the next level, anchor terms chosen from the new clustering results are set as the final term set for this taxonomy node.

The details are presented in the remaining of this section. Section 4.5 introduces anchor term selection method which is used multiple times across our framework. Sections 4.6.1 and 4.6.2 present how to learn term embedding from textual and motif contexts, respectively. Section 4.6.4 discusses the joint term embedding after we introduce our motif selection technique in Section 4.6.3. Finally, Section 4.7 shows how to allocate terms and documents into child taxonomy nodes.

4.5 ANCHOR TERM SELECTION

In order to provide more accurate initialization for the later instance-level motif selection, we first introduce our anchor term selection.

The goal of the anchor term selection is to find a concise, discriminative subset of terms from each cluster. It is a critical step for us to obtain clean semantics of a cluster, given that our vocabulary is large and noisy. For this very reason, we use anchor terms (1) as initialization for our instance-level motif selection module, in which they provide more accurate initial clustering information; (2) as input to the clustering algorithm, in order to find sub-topics under the current taxonomy node; and (3) as the final list of terms presented at each taxonomy node.

We formulate the anchor term selection as an unsupervised term ranking problem.

**Ranking Principles.** Given a specific taxonomy node, we define the anchor terms from the following criteria.

- **Popularity:** An anchor term should be popular enough at the given node. Very low frequency terms within a node do not contribute substantially to its semantics and so are not considered representative.
- **Discriminativeness:** An anchor term should be able to distinguish this node from its parent node and its sibling nodes. Discriminativeness is particularly critical in taxonomy scenarios, so analysts won’t be confused by two similar taxonomy nodes during
the navigation to find subsets of interest. Non-discriminative terms will appear in documents associated with many nodes and offer redundant and confusing information. For example, “extensive experiments” might be popular at both nodes about “data mining” and “database”, thus being non-discriminative.

- **Informativeness**: An anchor term should not be a stopword-like term. As the taxonomy construction goes deeper and deeper, some terms become less and less informative. For example, “data mining” is an informative term at the node representing the “computer science” field, but has much less information at the node focusing on “frequent pattern mining”.

Bearing these principles in minds, we design the following scoring functions accordingly.

**Popularity Score.** We denote the number of occurrences of the term $t$ in the document $d$ as $\text{tf}(t, d)$. As the documents are weighted, term frequency is weighted by the importance of the document. Given the document weights $w_{c,d}$, we define the popularity of the term $t$ at the node $c$ as

$$
\text{pop}(c, t) = \frac{\sum_{d \in D} w_{c,d} \cdot \text{tf}(t, d)}{\sum_{d \in D} w_{c,d} \cdot |d|}
$$

where $|d|$ represents the total number of terms in the document $d$. This formula captures the relative weighted term frequency of the term $t$ at the node $c$.

**Discriminativeness Score.** A discriminative term $t$ at the taxonomy node $c$ should have a significantly larger relative weighted term frequency at the node $c$ than that at its parent node $p_c$ or other sibling nodes $c'_1, c'_2, \ldots, c'_m$. Therefore, we define the following ratio to capture this intuition.

$$
\text{discriminative}(c, t) = \frac{\text{pop}_{c,t}}{\max\{\text{pop}_{p_c,t}, \max_{i=1}^m \text{pop}_{c'_i,t}\}}
$$

The larger $\text{discriminative}(c, t)$ should imply a better anchor term candidate. When $\text{discriminative}(c, t)$ is smaller than 1, it is unlikely that $t$ is a good choice of an anchor term at taxonomy node $c$.

**Informativeness Score.** Inverse document frequency (IDF) has been widely adopted in information retrieval to measure the informativeness of a term within a given corpus [117]. At each taxonomy node $c$, we calculate the weighted inverse document frequency as follows.

$$
\text{idf}(c, t) = \log \frac{\sum_{d \in D} w_{c,d}}{\sum_{d \in D} \mathbb{I}(t \in d) \cdot w_{c,d}}
$$

64
where $\mathbb{1}(t \in d)$ is a boolean indicator function about whether the term $t$ appears in the document $d$.

**Combined Anchor Score.** As an unsupervised ranking problem, we follow the previous comparative analysis work [118] and use a geometric mean to combine these three signals.

$$
\text{anchor\_score}(c, t) = \left( \text{pop}(c, t) \cdot \text{discriminative}(c, t) \cdot \text{idf}(c, t) \right)^{1/3}
$$

In summary, at each taxonomy node $c$, we will rank the terms based on the anchor scores and pick the top $K_t$ terms as anchor terms. We expect these anchor terms can express clear semantics of the topic at each node.

### 4.6 Joint Embedding from Textual and Motif Contexts

#### 4.6.1 Local Embedding from Text Data

In NetTaxo framework, term embedding is the key to discover sub-topic clusters at every taxonomy node.

Term embedding learning is typically conducted on the entire document collection [119, 120]. However, such learning paradigm faces a major drawback in topic taxonomy construction: the discriminative power of learned term embedding becomes limited at deep levels. For example, term embedding learned from all computer science papers shall be able to distinguish “machine learning”-related terms from terms in other research field. However, it may have difficulties in further discovering sub-topics under “machine learning”, as those “machine learning”-related terms are already quite close to each other. This problem will only get worse as we drill down further. Therefore, it is a necessity to condition the term embeddings on the current taxonomy node.

To this end, we follow previous work [88] and adopt the idea of local embedding [121] to learn term embedding from text data. The basic idea of local embedding is to fine-tune term embedding at each node according to its own associated (weighted) documents. Its effectiveness has been verified in [88] through ablation tests.

We use skip-gram with negative sampling (SGNS) [119] as our base embedding model. At each taxonomy node, we use local documents $\mathcal{D}_c$ instead of $\mathcal{D}$ for training. Similar to the original SGNS model, the objective is to maximize the probability of the local context given
a term in a document. The loss function to minimize is given by:

\[ L_{\text{text}} = \mathbb{E}_{d \sim P_D(D_c)} \left[ \sum_{t \in d} \sum_{t' \in C(t)} -P(t' \mid t) \right] \] (4.5)

\( C(t) \) stands for the set of terms within a context window of term \( t \). We sample documents according to the multinomial distribution \( P_D(D_c) \) parameterized by the document weights \( \{w_{c,d}\} \) under the current taxonomy node. Therefore, our loss function slightly differs from the ones in the previous work [88] as well as the original local embedding work [121].

4.6.2 Motif Instances as Term Contexts

We generalize the distributional hypothesis, which is fundamental in word embedding, to network by using motif instances. In text data, every word within sliding windows of a term is regarded as a part of its contexts. Similarly, a term’s motif context is characterized by the set of motif instances, which the term can match based on network structures and the provided motif patterns. The network version of distributional hypothesis therefore becomes: terms with similar motif contexts are similar.

Now we can generalize the SGNS embedding model to incorporate motif context. Specifically, we use each term to predict its motif context, generating the following loss term.

\[ L_{\text{motif}} = \mathbb{E}_{d \sim P_D(D_c)} \left[ \sum_{t \in d} \mathbb{E}_{m \sim \hat{M}_c(t)} - \log P(m \mid t) \right] \] (4.6)

where \( \hat{M}_c(t) \) is the associated motif instances of term \( t \). We will describe how to select \( \hat{M}_c \) in section 4.6.3.

The probabilities are approximated with negative sampling [119].

\[ \log P(m \mid t) = \log \sigma(\mathbf{r}_m^T \mathbf{u}_t) - \mathbb{E}_{m \sim P_{\text{neg}}(m)} \left[ \log \sigma(-\mathbf{r}_j^T \mathbf{u}_t) \right] \] (4.7)

where \( \mathbf{r} \) and \( \mathbf{u} \) are embedding vectors of motif instances and terms and \( P_{\text{neg}}(m) \) is the negative sampling distribution.

In this way, term embedding can be also derived from network structures given the user-provided motif patterns.
4.6.3 Instance-Level Motif Selection

So far we have already shown how to learn term embedding separately from text and motif using local corpus. Trivially putting them together, however, gives sub-optimal performance based on our observation. As discussed before, motif instances should be weighed accordingly at each taxonomy node during the construction process. Specifically, based on anchor terms selected from initial clusters, we further narrow down a set of useful motif instances. This instance-level motif selection step is designed to make the collaboration between text and network more effective.

We identify two principles for instance-level motif selection:

• **Importance**: The motif instance should be associated with a set of important terms, providing useful information for term embedding learning.

• **Concentration**: The motif instance should be concentrated on one or a small number of sub-topics under the current taxonomy node, thereby including it will help us better separate sub-topics.

We realize these two principles by applying authority ranking \[109\] upon the motif context graph.

The motif context graph at the taxonomy node \(c\) is a bipartite graph \(G^M_c = (T_c, M_c, W)\), where \(T_c\) is the terms under the current taxonomy node and \(M_c\) is the set of motif instances. We use the notation \(G^M_c\) to avoid the ambiguity of mixing this graph with the network structure \(G\). Note that we exclude motif instances which doesn’t include any term or document under the current taxonomy node. The bipartite graph connects each term to the motif instances it occurs in. The weight matrix \(W \in \mathbb{R}^{|T_c| \times |M_c|}\) describes the number of occurrences of term \(t\) in each motif instance \(m\) (i.e., \(W_{t,m}\)).

We apply authority ranking to obtain importance scores between each motif instance and each cluster. In the ranking process, we maintain two matrices \(I_T \in \mathbb{R}^{|T_c| \times n}\) and \(I_M \in \mathbb{R}^{|M_c| \times n}\) to store the importance scores of terms and motif instances. Each row of the matrix denotes the importance scores of a specific term (or a motif instance) under all \(n\) clusters. As initialization, we set \(I_T^{(0)}(t,k) = 1/K_t\) for all anchor terms in all clusters and zero for all other terms. This is based on the assumption that all anchor terms are important in the first place. The authority ranking is an iterative importance propagation process. Specifically, in each iteration,

\[
I_M^{(t)} \leftarrow \bar{W}^T I_T^{(t-1)}, \quad I_T^{(t)} \leftarrow \bar{W} I_M^{(t)}
\]  

\(\bar{W} = \frac{1}{2} D_r^{-1/2} W D_c^{-1/2}\) is the normalized weight matrix with row degree \(D_r\) and column degree \(D_c\) matrices. The iterative process can be repeated to a max iteration number or until convergence. In practice, we found that 5 iterations are enough to achieve good results.
For each motif instance \( m \), we take the mean of its importance score across different clusters as the overall importance.

\[
\text{importance}(m) = \text{mean}(I_M(m, \cdot))
\] (4.9)

Moreover, with the importance scores on different clusters, we can measure the concentration of a motif instance \( m \) based on entropy.

\[
\text{concentration}(m) = 1 - \frac{1}{\log n} \sum_{i=1}^{n} \tilde{I}_M(m, i) \log \tilde{I}_M(m, i)
\] (4.10)

We use normalized entropy here to keep its range in 0 to 1. \( \tilde{I}_M \) denotes \( I_M \) after row normalization.

Finally, we define the final score of a motif instance \( m \) as

\[
\text{motif\_score}(m) = (\text{importance}(m) \cdot \text{concentration}(m))^{1/2}
\] (4.11)

We rank all motif instances based on their final scores, and select a subset \( \tilde{M}_c \) of the instances ranked in the top \( K_m \) percent. Note that, the motif instance ranking is across all motif patterns. Therefore, we are implicitly selecting motif patterns by pruning most of instances from uninformative motif patterns.

4.6.4 Joint Embedding Training

At each taxonomy node \( c \), given the local corpus \( D_c \) and locally selected motif instances \( \tilde{M}_c \), we refine term embedding by joint embedding training of text and motif instances. Specifically, putting text and motif together, we minimize the joint loss function:

\[
\mathcal{L} = \lambda \mathcal{L}_{\text{text}} + (1 - \lambda) \mathcal{L}_{\text{motif}}
\] (4.12)

We use \( \lambda \) to balance text and motif losses. In our implementation, we optimize the loss function with stochastic gradient descent, and approximate the expectations in previous equations using sampling.
Table 4.1: Dataset Statistics. Motifs patterns in DBLP-5 and Yelp-5 datasets are visualized in Example 4.1.

<table>
<thead>
<tr>
<th></th>
<th>#doc</th>
<th>#term</th>
<th>#node</th>
<th>#edge</th>
<th>#motif</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBLP-5</td>
<td>79,896</td>
<td>26,684</td>
<td>182,290</td>
<td>1,897,226</td>
<td>5</td>
</tr>
<tr>
<td>Yelp-5</td>
<td>1,308,371</td>
<td>74,951</td>
<td>1,760,025</td>
<td>6,809,152</td>
<td>4</td>
</tr>
</tbody>
</table>

4.7 TERM AND DOCUMENT ALLOCATION

With the joint embedding trained on text and motif instances, we are ready to allocate terms and documents into children nodes. In principle, our method is flexible in the choice of clustering method. Consider that cosine similarity between term embedding has demonstrated its effectiveness in term similarity search [119], we apply vMF mixture clustering [122] in NetTaxo. It is a classical, effective soft clustering method on the unit hyper-sphere. Since the constructed topic taxonomy rarely changes, we leave the choice of $k$, the number of topics, to human experts.

It is worth noting that we fit the vMF distributions only on anchor terms of the current taxonomy node. The rationale is that the automatically extracted term vocabulary is often noisy, while anchor terms selected from comparative analysis are much cleaner, which makes the clustering more accurate. After fitting the vMF mixture model, each cluster is represented by a vMF distribution in the embedding space. We then use these distributions to estimate the clustering probability of each term in $\mathcal{T}_c$. Finally, we allocate terms to children clusters.

For documents in $D_c$, we estimate their clustering probability by aggregating clustering probability from their connected terms. This process is the same as that in [88]. The aggregated probabilities of a document, multiplied by its current weight, will be the weights of the document on the next level.

4.8 EXPERIMENTS

In this section, we first introduce the experimental settings, including datasets, compared methods, and evaluation metrics. We then present quantitative evaluation results. In the end, we showcase parts of the constructed topic taxonomies as well as several interesting findings.
### 4.8.1 Datasets

We conduct our experiments on two real-world document collections: computer science papers in DBLP and business reviews in Yelp. The statistics about the two datasets are shown in Table 4.1. Details about the two datasets are as below.

- **DBLP-5.** The first document collection is from the AMiner dataset about computer science papers\(^1\). We select five closely-related research areas: (1) data mining, (2) database, (3) machine learning, (4) computer vision, and (5) natural language processing. From these five areas, 79,896 papers are selected, containing 26,684 distinct terms. The network contains node types of author, venue, year, paper, and term (as available in this DBLP dataset). We augment the network by adding “year range” nodes, each representing a five consecutive years (e.g., 2010-2014). The text data, i.e., title and abstract, is associated with each paper node. The edges describe author–paper, venue–paper, year–paper, year range–paper, and term–paper relations. Note that, previous methods [88, 106] choose five areas from this dataset too, for example in [88], information retrieval, computer vision, robotics, security & network, and machine learning. In contrast, our chosen five areas are more closely related to each other, thus being more challenging.

- **Yelp-5.** The second document collection is from the Yelp Dataset Challenge\(^2\). Since some baselines are too slow if we use the full dataset, we have to choose a subset of these reviews. Particularly, we choose the most popular state (i.e., Arizona) and the top-5 popular business categories (i.e., (1) automotive, (2) beauty & spas, (3) hotels & travel, (4) restaurants, and (5) shopping). We also remove rare businesses with less than 50 reviews. As a result, we obtain 1,308,371 reviews in total and extract 74,951 terms from them. We build the network using nodes of business, user, review, and term and edges of business–review, user–review, and term–review, as they are available in the meta-data. The text data, i.e., review comments, is associated with each review node.

We present all motif patterns used in DBLP-5 and Yelp-5 datasets in Example 4.1.

### 4.8.2 Compared Methods

We compare our proposed methods with different types of topic taxonomy construction methods: (1) using text data, (2) using network data, and (3) using both text and network

---

1. https://aminer.org/citation
data. The details are listed below.

- **HPAM++** is a method enhanced by us from the original Hierarchical Pachinko Allocation Model (HPAM) [104]. HPAM is a state-of-the-art hierarchical topic model built upon the Pachinko Allocation Model *using text data*. Although it was designed to work on all unigrams, to make the comparison more fair, we improve the HPAM by focusing on only very high-quality phrases. Also, we set the topic numbers at different levels as the same as the numbers of clusters in NetTaxo. We have tested our enhanced Hierarchical Latent Dirichlet Allocation (HLDA) model [103] and its performance is quite similar to HPAM++. Therefore, we only present the results of HPAM++ here.

- **TaxoGen** [88] is the state-of-the-art topic taxonomy construction method *using text data*. As demonstrated in its paper, it beats many strong baselines, such as hierarchical topic models [103, 104, 105, 106, 107]. It utilizes the same local embedding idea as our model, but ignores network structures.

- **CATHYHIN++** is a method enhanced by us from the original CATHYHIN [106] method. CATHYHIN [106] is a topic taxonomy construction method *using network data*. It treats unigrams as nodes and attempted to mine terms (i.e., phrases) and clusters simultaneously. Its performance is limited due to (1) the poor phrase quality compared to the state-of-the-art method [5] and (2) the poor term clustering results compared to methods that use the term embedding technique. To make the comparison more fair, we improve the CATHYHIN by adding only very high-quality phrases.

- **HClusEmbed** is a baseline method that we proposed *using both text and network data*. It is a straightforward solution to combine the term embedding technique with the network structure. Specifically, we first learn term embedding vectors from text using word2vec [119] and network using LINE [123] separately, where every embedding vector has a dimension of 300. And then, we concatenate the two vectors for each term, and then apply hierarchical spherical k-Means algorithm. We name this method as hierarchical topic clustering based on term and node embedding, and therefore denote it as HClusEmbed.

We denote our proposed method as NetTaxo. To demonstrate the necessity and effectiveness of our proposed motif instance selection, we introduce an ablated version of NetTaxo without this step, denoted as NetTaxo w/o Selection.

Note that, in order to conduct a fair comparison, the same set of terms are used across different methods. They are the extracted from raw texts by the state-of-the-art distant supervised phrase mining method [5].
4.8.3 Parameter Setting

The number of mixtures $k$ for vMF mixture clustering is manually selected by incrementally increasing $k$ by 1 in the range of $[3, 6]$ until coherent clusters are observed. We set $k = 5$ for the top level and $k = 4$ for the second level of the taxonomy in both the DBLP and Yelp dataset. In TaxoGen [88], this number is set to 5 for all levels, which is not far from our observation. Note that this parameter will only need to set once for a given dataset, so this process will not put a large burden on humans. For anchor term selection, we use $K_t = 50$ for each cluster. For motif selection, we keep top $K_m = 10\%$ of motif instances.

4.8.4 Evaluation Tasks & Metrics

Systematic evaluation of the constructed topic taxonomy has long been a very challenging task. Inspired by the state-of-the-art work on topic taxonomy construction [106, 88] and recent work on topic modeling [124, 125], we design a set of tasks for human evaluation. For each dataset, we recruited 10 in-domain human experts. In their annotation process, they were encouraged to use search engines (e.g., Google) to better understand unfamiliar terms.

We identify the following aspects for judging the taxonomy quality, and then design three evaluation tasks accordingly.

- **Coherence.** Within each node in the taxonomy, the terms should be able to form a semantically coherent topic. Similar to previous topic model evaluations [124, 125], we present the top-5 terms to human annotators from the same taxonomy node. Annotators are asked to first judge whether these terms form an interpretable topic. If not, all five terms at this node are automatically labeled as irrelevant. Otherwise, annotators are then asked to identify specific terms that are relevant to this topic. We define the coherence measure as the ratio of the number of relevant terms over the total number of presented terms.

- **Exclusive Siblings.** Besides the coherence, each taxonomy node should be distinguishable from its sibling nodes. Following previous taxonomy construction methods [106, 88], we perform the term intrusion test. Specifically, for each node, we collect its top-5 terms, and then randomly mix in an intruder term from the top-5 terms of its sibling nodes. We present the 6 terms in a random order and ask human annotators to identify the only intruder term. The more coherent and distinctive the topics are, the easier it is for human to spot intruder terms. We define the sibling exclusiveness as the successful identification ratio in this test.

- **Quality Parent-Child Relations.** Each taxonomy node should be an appropriate
sub-topic of its parent node. Considering the huge vocabulary size, it is difficult to enumerate all children terms of a given topic, and further evaluate the relation quality. We instead use a sampling-based method for evaluation. Specifically, between two adjacent levels in a taxonomy, we first sample a child term $t$ from lower-level nodes, and present $t$ together with all upper-level (i.e., parent-level) nodes. Each upper-level node is visualized using its top-10 terms. We ask human annotators to mark all reasonable parent nodes of the child term $t$, which is denoted as $\hat{\mathcal{P}}(t)$. We merge the parent nodes of term $t$ that identified by the model into a set $\mathcal{P}^*(t)$. Precision, recall, and F1 are employed to evaluate $\hat{\mathcal{P}}(t)$ against $\mathcal{P}^*(t)$ by treating all sampled together. Formally, we have

\[
\text{Precision} = \frac{\sum_t |\hat{\mathcal{P}}(t) \cap \mathcal{P}^*(t)|}{\sum_t |\mathcal{P}^*(t)|} \quad (4.13)
\]

\[
\text{Recall} = \frac{\sum_t |\hat{\mathcal{P}}(t) \cap \mathcal{P}^*(t)|}{\sum_t |\hat{\mathcal{P}}(t)|} \quad (4.14)
\]

and F1 is defined as their harmonic mean.

A quality topic taxonomy should have high scores in all the three evaluation tasks.

**Annotation Details.** First of all, it is worth mentioning that we mix the results from different methods together and shuffle them randomly before sending them to annotators. The annotators will not be aware of the method from which the results are produced.

Second, in order to avoid bias during the annotation, we first ask the annotators to do the *Exclusive Siblings* task, then the *Parent-Child Relations* task, and finally the *Coherence* task. So the annotator will not have any prior knowledge about which terms are in the same taxonomy node in the first two tasks.

In all tasks, we observe that annotators have inter-annotator agreements of more than 90%. The scores presented in the experiments are therefore all averaged across different annotators.

---

**Table 4.2: Quantitative Evaluations. Scores are averaged over 10 annotators.**

<table>
<thead>
<tr>
<th></th>
<th>DBLP-5</th>
<th></th>
<th>Yelp-5</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Coherence</td>
<td>Sibling</td>
<td>Parent-Child Relations</td>
<td>Coherence</td>
</tr>
<tr>
<td>Measure</td>
<td>Exclusiveness</td>
<td>Precision</td>
<td>Recall</td>
<td>F1</td>
</tr>
<tr>
<td>HPAM++</td>
<td>0.796</td>
<td>0.680</td>
<td>0.348</td>
<td>0.451</td>
</tr>
<tr>
<td>TaxoGen</td>
<td>0.840</td>
<td>0.740</td>
<td>0.780</td>
<td>0.713</td>
</tr>
<tr>
<td>CATHYHIN++</td>
<td>0.880</td>
<td>0.533</td>
<td>0.850</td>
<td>0.744</td>
</tr>
<tr>
<td>HClusEmbed</td>
<td>0.624</td>
<td>0.420</td>
<td>0.525</td>
<td>0.409</td>
</tr>
<tr>
<td>NetTaxo w/o Selection</td>
<td>0.908</td>
<td>0.680</td>
<td>0.895</td>
<td>0.808</td>
</tr>
<tr>
<td>NetTaxo</td>
<td>0.912</td>
<td>0.880</td>
<td>0.898</td>
<td>0.810</td>
</tr>
</tbody>
</table>
4.8.5 Quantitative Evaluation

In this section, we discuss the quantitative evaluation results of different methods on the two datasets. The results are summarized in Table 4.2. Overall, the topic taxonomy constructed by \textbf{NetTaxo} has demonstrated its significant advantage over taxonomies constructed by other methods, in all three evaluated aspects.

The two datasets have slightly different properties in that text in the DBLP dataset is written in a more formal style and thus consists of cleaner terms while Yelp reviews often contain colloquial language. In terms of the underlying taxonomy, the Yelp taxonomy spans from very high-level distinctions (e.g., \textit{auto repairs} vs. \textit{restaurants}) to subtle distinctions (e.g., fusion dishes should belong to multiple nodes while common products do not fit into any node). The DBLP taxonomy has much fewer cases of ambiguity. This leads to closer but generally lower scores on the Yelp dataset for most metrics.

Within two methods using text data only, \textbf{TaxoGen} outperforms \textbf{HPAM}++ in all evaluated aspects. Compared with \textbf{TaxoGen}, \textbf{NetTaxo} improves most on the identification of parent-child relations. The network information provides a better overview of the hierarchical topic structure, whereas parent term and child terms often share the same context in documents. This shows that the network structure truly provides complementary information to text.

Compared with \textbf{CATHYHIN}++, \textbf{NetTaxo} shows significant improvements in sibling exclusiveness and coherence. By initializing clusters using term embedding, we are able to better capture semantically similar terms for creating coherence clusters. The increase in sibling exclusiveness can be credited to the comparative analysis component, which puts sibling nodes under contrast to discover anchor nodes. Both of these components are only available with text data, which \textbf{CATHYHIN}++ does not leverage.

In short, \textbf{NetTaxo} outperforms \textbf{TaxoGen} and \textbf{CATHYHIN}++ in all metrics, demonstrating that text and network information are able to enhance each other.

\textbf{HClusEmbed} takes the same input as \textbf{NetTaxo}, but performs very poor among the baselines. This shows that the trivial combination of word and network embedding is not enough to generate a high quality taxonomy. In \textbf{HClusEmbed}, term embedding aim to preserve semantic similarity while network embedding aim to preserve node proximity, both may not directly contribute to a better taxonomy. Moreover, checking the results of \textbf{NetTaxo w/o Selection}, one can observe that only a careful selection of the information from network structures can lead to the performance gains. This is more significant on the Yelp-5 dataset, as the network information is much more noisy. \textbf{NetTaxo} is carefully designed to select the most relevant motif contexts from network structures, and then incorporate them into a joint term embedding learning to further improve the quality of constructed taxonomy.
Figure 4.6: The Topic Taxonomy Constructed by NetTaxo on the DBLP Dataset. Due to the space limit, it is partially expanded. Each node is visualized as a rectangle block and its top-10 anchor terms. Arrows go from parent nodes to child nodes. In addition, at the first level, for each motif pattern, we show the percentage of selected instances over all instances of the motif.

These comparisons further confirm the importance and effectiveness of our proposed motif selection process.

4.8.6 Constructed Topic Taxonomies

After the quantitative comparison, we present some case studies on both datasets for a closer look at the topic taxonomy constructed by our proposed NetTaxo.

DBLP Taxonomy. We plot the final topic taxonomy in Figure 4.6. Due to the space limit, we only present the five nodes at the first level and expand two of them into the second level.

Looking at the top level topics, one can easily recognize the topics of the five nodes from the left to right as: (1) natural language processing, (2) machine learning, (3) database, (4) computer vision, and (5) data mining. These are exactly the five areas used in preparing the DBLP dataset.

In addition, we present the selected motif instance percentage for each motif pattern at this top level in Figure 4.6. The results are intuitive by putting more emphasis in venue-related motif patterns as well as the author-pair motif pattern. While we are conducting the motif instance-level selection, it actually selects the motif patterns in an implicit way too.

We then inspect the second-level results. Under the node about natural language processing, we can see four clear sub-topics: (1) parsing, (2) information extraction, (3) language models, and (4) natural language generation.
& grammar, and (4) machine translation. Under the node about data mining, we can find (1) social network analysis, (2) web mining and search, (3) frequent pattern/association rule mining, and (4) clustering.

Yelp Taxonomy. In Figure 4.7, we present all four taxonomy nodes under the taxonomy node of “Asian Food” topic in our constructed topic taxonomy on the Yelp dataset. Top-10 terms are presented at each node. While “Asian Food” is already a relatively fine-grained topic, NetTaxo successfully recovers its sub-topics: Thai cuisine, Japanese cuisine, Chinese cuisine, and Other Asian (e.g., Indian, Mexican-Chinese Fusion, ...) cuisines. The first three sub-topics are quite clear, while the fourth one is a little vague. Remember that we set \( k = 4 \) here. So it makes sense to have an “other” sub-topic. At the first glance, “Jade Red Chicken” and “Emerald Chicken” look like Chinese dishes, and “Jerk Fried Rice” sounds like something from the Caribbean area. However, if one searches “Jade Red Chicken” in Google, a popular restaurant in Arizona named “Chino Bandido” pops up at the first place. It offers Mexican-Chinese Fusion dishes and these three dishes are strongly recommended by Yelp reviewers\(^3\).

4.8.7 Effects of Instance-Level Motif Selection

Besides the final taxonomy quality, we’re also interested in how the instance-level motif selection mechanism works at different taxonomy nodes. We visualize top motif instances selected by our method on the DBLP dataset. Figure 4.8 shows two motif patterns and their top instances at two taxonomy nodes, one from the first level and the other from the second

\(^3\)https://www.yelp.com/menu/chino-bandido-chandler/item/jade-red-chicken
Figure 4.8: Top motif instances selected by NetTaxo at different taxonomy nodes. For venue + year range motif pattern, we merge consecutive instances for readability purpose if they are from the same venue and cover contiguous years. Three highlighted motif instances will be further elaborated with their most frequent terms in Figure 4.9.

Figure 4.9: Most frequent terms linked to the three highlighted motif instances in Figure 4.8. Note that the frequency is calculated based on the weighted documents associated with the taxonomy node about NLP. Best viewed in color.

level. Taking a closer look at three specific motif instances, we show most frequent terms linked to these motif instances in Figure 4.9.

On the first level, our goal is to identify major research fields, i.e., separating the 5 research areas in this dataset. From co-authorship motif pattern, we observe pairs of database researchers who share lots of research papers. The top-2 instances are all professors working in the same research group at the same university. From venue-and-year-range motif pattern, one can find many computer vision and database conferences. The reason for these motif instances to rank high is because database and computer vision are two relatively concentrated research areas, compared to machine learning, data mining, and natural language processing (NLP) which have more interconnections. Besides, professors and venues involved in the top-ranked instances are all highly reputed.

On the second level, the goal becomes more challenging — distinguishing research sub-areas. We use the NLP taxonomy node as an example. The co-authorship motif instances give us some less-known researchers. Therefore, we picked two of co-author pairs, sampled and visualized their associated terms from the motif context graph, shown in Figure 4.9. One can easily observe that these author groups work on relatively concentrated sub-topics...
under NLP, i.e., sentiment analysis and machine translation, respectively. From venue-and-year-range motif instances, we can see major NLP conferences in their early years, and data mining conferences in recent years. This is also quite interesting but explainable, as NLP conferences have a narrower scope in their early years, while the data mining community, as it evolves, has more overlaps with the NLP community recently. Specifically, we show most frequent terms linked to the motif instance “CIKM 2010-2014” in Figure 4.9, where we observe many NLP sub-topics such as “question answering” and “information extraction”. These topics are also studied by information retrieval and data mining researchers recently.

Overall, from the empirical observations, we can verify that our instance-level motif selection is effective.

4.9 SUMMARY

In this paper, towards automatic topic taxonomy construction, we propose a novel hierarchical term embedding and clustering framework NetTaxo, which consumes a text-rich network as the input. Through a careful selection of motif contexts, NetTaxo learns term embedding jointly from the text data collaborate with the most helpful network structures. To consolidate the foundation of such selection, we further design a method to choose anchor terms from the initial clusters based on text data only. Extensive experiments on two datasets demonstrate the superiority of our framework compared with baselines. Ablation experiments confirm the necessity and effectiveness of our proposed instance-level motif selection. Case studies illustrate the quality of our constructed taxonomy.

For future work, we would like to further improve NetTaxo in the following aspects. First, we would like to develop a more principled solution to determine the number of sub-topics at each taxonomy node. Second, incorporating user-provided seed examples of the desired taxonomy in construction process could be a promising and practically useful direction to pursue. Last but not least, we are interested in integrating our constructed taxonomy into downstream applications, such as recommender systems and question answering tasks.
CHAPTER 5: AUTONET DEMO SYSTEM

5.1 OVERVIEW OF AUTONET DEMO SYSTEM

The majority of the massive volume of real world data consists of unstructured or loosely structured text, ranging from news to social media, web contents, scientific papers, government documents, and business contracts. The sheer size of such data and the fast pace of new data generation make many existing approaches unscaleable and infeasible, due to their reliance on heavy human annotation and curation at the extraction of named entities and their relationships as well as the construction of knowledge graphs. Therefore, automated structure discovery and construction from massive text corpora have become an active research area in the fields of data mining, machine learning, and natural language processing.

We propose a novel and principled data-driven approach for automatic knowledge discovery in massive, unstructured, and noisy text corpora by constructing high-quality and structured heterogeneous information networks (HINs), in a distantly-supervised manner. Note that the HINs that we propose to construct provide stronger typed and structural information than typical designs of knowledge graphs and thus endowing stronger power for mining and inference as shown in [126]. Moreover, our proposed approach is general, extensible to text corpora in multiple natural languages and across multiple domains. Therefore, instead of common-sense knowledge, the HINs will be directly constructed from the given corpus, and thus can uncover the domain-specific knowledge.

5.2 AUTONET SYSTEM DESIGN

Fig. 5.1 shows the workflow for AutoNet depicting its two main innovative features: 1) model learning and network construction, and 2) network exploration and construction on the fly.

5.2.1 Model Learning and Network Construction

AutoNet only requires massive unlabeled texts and existing knowledge bases (KBs), and then learns a series of models, without additional human effort. Such models mine structures (i.e., entities and relations) from text using minimal language- or domain- dependent features. Therefore, the user can easily adapt AutoNet to his/her own domain/language by providing new corpus/KB. Finally, AutoNet constructs a large-scale HIN based on mined
5.2.2 Network Exploration and Construction on the Fly

**AutoNet** will retrieve related nodes and edges, if any, from the large-scale HIN and construct new HINs on the fly from user-provided documents with a similar network construction process guided by the saved models. Also, a user can explore the subnetwork by keywords. The built indices will facilitate efficient selection. An interactive network visualizer enables effective explorations. The node color reflects its type, the node size shows its popularity, and the link thickness means its frequency. As our relation extractor ReMine [14] provides relational phrases, we summarize every relation between two entities using a word cloud of all its relational phrases weighted by their frequencies. Moreover, for each relation, **AutoNet** presents its grounded documents to the user for further investigation.

To our best knowledge, **AutoNet** is the first system that can construct HINs in the user-specified domain and language. Specifically, it has the following three innovative components.

1. **Phrase Mining.** We have successfully developed two novel phrase mining methods, SegPhrase [4] and AutoPhrase [5]. They can automatically extract high-quality phrases from domain-specific text corpora written in different languages under light supervision or distant supervision. It’s worth mentioning that our phrase mining tools have received Yelp Dataset Challenge Grand Prize\(^1\) and reported by TripAdvisor in their business usage\(^2\).

2. **Entity Recognition.** At the corpus-level, we have recently developed ClusType [62] and PLE [127], two distantly-supervised models for coarse-grained and fine-grained

---

\(^1\)https://www.yelp.com/dataset/challenge/winners
\(^2\)http://engineering.tripadvisor.com/using-nlp-to-find-interesting-collections-of-hotels/
typings at the corpus-level. At the sentence-level, our LM-LSTM-CRF model has achieved the state-of-the-art on benchmark datasets under the supervised setting without any other external resources [6]. Going beyond, we have developed another distantly supervised sentence-level entity recognition model, AutoNER [8].

3. Relation Extraction and Attribute Discovery. ReMine [14] is a novel distantly supervised open-domain information extraction (Open IE) method. It can extract high-confidence relational phrases from domain-specific texts in an end-to-end manner, therefore, it receives WWW’18 best poster award honorable mentioning. Built upon phrase mining methods, we have developed MetaPAD [15] to extract attribute names and values with light effort.

5.3 PREVIOUS EFFORTS AND LIMITATIONS

Domain-specific search engines, such as PubMed, using keywords and Medical Subject Headings (MeSH) terms might not work well on capturing cross-document entity relations or identifying publications related to these relations. Life-iNet is a recently proposed network-based knowledge exploration system [128], however, it cannot support online network construction and also can only explore pre-defined relations.

5.4 DEMO RESULTS AND VIDEO

Within a few hours, AutoNet can construct a HIN of more than 64 million nodes and 186 million edges based on 2.93 million Cancer-related PubMed papers and the MeSH database, or a HIN of more than 40 thousand nodes and 110 thousand edges based on 2.77 million computer science paper abstracts and Wikipedia.

Demo video of AutoNet system is available at YouTube: https://www.youtube.com/watch?v=tdtBigWq_vo&feature=youtu.be. It mainly demonstrates the functions of “Network Exploration” and “Construction on the Fly”.

5.5 OUR OPEN-SOURCE REPOSITORIES

Our key methods used in this system are all open-sourced on GitHub as follows and have received over 1500 stars, including phrase mining, entity typing, relation extraction, and attribute discovery. Here is a list of our open-sourced tools, related to this thesis.

• Phrase Mining
- AutoPhrase: https://github.com/shangjingbo1226/AutoPhrase
- SegPhrase: https://github.com/shangjingbo1226/SegPhrase

**Named Entity Recognition**
- AutoNER: https://github.com/shangjingbo1226/AutoNER
- CrossWeigh: https://github.com/ZihanWangKi/CrossWeigh
- Raw-to-End: https://github.com/LiyuanLucasLiu/Raw-to-End

**Relation Extraction and Attribute Discovery**
- ReMine: https://github.com/GentleZhu/ReMine
- MetaPAD: https://github.com/mjiang89/MetaPAD
CHAPTER 6: CONCLUSIONS, VISIONS, AND DISCUSSIONS

6.1 CONCLUSIONS

This thesis has demonstrated a feasible path of turning unstructured text data into structured heterogeneous information networks from which actionable knowledge can be generated based on the user’s need. The whole process is designed to be “automatic”. The key philosophy of “automatic” here refers to achieving reasonably great accuracy without introducing additional human effort, in both feature engineering and label annotations. Representation learning techniques, including both embedding and deep neural networks, free human experts from handcrafting features. Distant supervision, leveraging existing, public knowledge bases, provides a chance to get rid of manually curated training data. We put together these two ideas in the scenarios of information extraction and knowledge discovery and develop the AutoNet framework, which automated the text understanding process and organized unstructured text information into structured knowledge.

6.2 VISIONS FOR FUTURE

My long-term goal is to create general data-driven methods to transform text data of various kinds into structured databases of human knowledge. I am passionate about enabling powerful machines to act on such automatically transformed knowledge to improve human productivity in various real-world applications. I am also excited about applying my AutoNet techniques to construct networks from the scientific literature of various disciplines, and then discovering beneficial insights for multi-disciplinary scientific research, such as life sciences, public health, social science, cognitive science, environmental science, and economics.

I plan to continue my research along the path of corpus-to-network-to-knowledge to discover principles, propose methodologies, and design scalable solutions. My mixed research background places me in a unique position for solving this challenging problem: my experience of text mining and information network mining assists me in turning unstructured text data into structured networks, and my experience of machine learning and competitive programming competitions enables me to build efficient, scalable computational systems to analyze the constructed networks.

Here are a few future research projects that I am thrilled to explore.
6.2.1 Knowledge Enrichment in Constructed Networks

In the current AutoNet framework, edges between two entities are described by relational phrases and weighted by their frequency in the text corpus. Such a representation has several limitations: (1) relational phrases are not grouped into synonym sets yet (e.g., treat and cure), (2) frequency-based edge weight cannot reflect the uncertainty (e.g., drug A treats disease B with 75% success rate), (3) Conditions (e.g., if the patient is under 50 years old) are usually missed in relational phrases, and (4) complex relations involve more than two entities (e.g., protein localization relation) cannot be modeled in HINs. Therefore, I propose to (1) explore and develop synonym word/phrase clustering algorithms; (2) adapt my data-driven sentiment analysis method to identify and quantify uncertainty, negation, intensifier, and diminisher words/phrases; (3) extend my MetaPAD algorithm to identify patterns for “condition descriptions” (e.g., “... [with age ...]...”) and attach the mined conditions to edges for further analysis; and (4) generalize HINs in the AutoNet framework to hypergraphs.

6.2.2 Explanatory Embedding in Text-Rich Graph Mining

Most time, only network structure information of the constructed HINs in the AutoNet framework is leveraged in the downstream tasks (e.g., node classification and link prediction). Such a process ignores the nature of the constructed network – it is in fact closely associated with texts. For example, each node may have definitions in the corpus, and each link may have its semantics described by several phrases. Therefore, I plan to explore and answer the following two questions: (1) how to aid the node representation learning by incorporating texts like definitions? The enhanced representations can later benefit a wide spectrum of applications; and (2) can we develop a “decoder” to interpret the learned representations by “translating” the vectors back to texts (e.g., keywords or definition sentences) in natural language? This can improve the interpretability of node embedding dramatically.

6.2.3 Next-Generation Literature Search System

During scientific research, experts can only read a small subset of what is published in their fields, and are often unaware of developments in related fields. Existing literature search systems (e.g., ACM Digital Library, Google Scholar, and Semantic Scholar) can only present a ranked list of result documents. Can we present a comprehensive summary concisely and efficiently to improve the productivity of researchers? I am interested in: (1) building hierarchical network summaries based on machine-constructed networks from a set of documents;
and (2) forming candidate scientific hypotheses (for further in-depth examination) by doing knowledge inference. Integrating these two modules can lead to a transformative literature search system.

6.2.4 Fusing Text Mining and Physical Sensing

Our physical world constantly produces data of various types that can be collected by physical sensors (e.g., GPS sensors in smartphones). I am interested in the “fusion” of physical sensor data and unstructured text data. The physically sensed data can potentially lead to a better understanding of the texts (e.g., spatial-temporal information of a tweet helps to disambiguate the entities mentioned in it). Meanwhile, the text data can help decision-makers better understand their physical environment (e.g., summarizing tweets with a certain event tag helps to understand the event details).
REFERENCES


[115] J. Shang, M. Qu, J. Liu, L. M. Kaplan, J. Han, and J. Peng, “Meta-path guided
embedding for similarity search in large-scale heterogeneous information networks,”


