7/10 2020 - 79 pages - 2021.10.13 shorter version 10 pages
- Containing more implementation details
- good writing style / format for master thesis

Financial Knowledge Graph Construction

by

Sarah Elhammadi

B.Sc., Alexandria University, 2011
M.Sc., Alexandria University, 2018

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

in

The Faculty of Graduate and Postdoctoral Studies

(Computer Science)

THE UNIVERSITY OF BRITISH COLUMBIA

(Vancouver)

July 2020

The following individuals certify that they have read, and recommend to the Faculty of Graduate and Postdoctoral Studies for acceptance, the thesis entitled:

Financial Knowledge Graph Construction

submitted by **Sarah Elhammadi** in partial fulfillment of the requirements for the degree of **Master of Science** in **Computer Science**.

Examining Committee:

Laks V.S. Lakshmanan, Professor, Computer Science, UBC
Supervisor

Raymond Ng, Professor, Computer Science, UBC
Co-supervisor

Abstract

The proliferation of financial news sources reporting on companies, markets, currencies and stocks presents an opportunity for strategic decision making by mining data with the goal of extracting structured representations about financial entities and their inter-relations. These representations can be conveniently stored as (subject, predicate, object) triples in a knowledge graph that can be used drive new insights through answering complex queries using high level declarative languages. Towards this goal, we develop a high precision knowledge extraction pipeline tailored for the financial domain. This pipeline combines multiple information extraction techniques with a financial dictionary that we built, all working together to produce over 342,000 *compact* extractions from over 288,000 financial news articles, with a precision of 78% at the top-100 extractions. These extractions are stored in a knowledge graph readily available for use in downstream applications. Our pipeline outperforms existing work in terms of precision, the total number of extractions and the coverage of financial predicates.

TransE h + r ≈ t

linear / rotational transformations

one-to-one mapping

Lay Summary

There is significant interest in building Knowledge Graphs from unstructured textual sources to power downstream applications such as Question Answering and computational fact checking. This thesis presents a high precision knowledge extraction pipeline tailored for the financial news domain. The extracted structured representations are stored in a knowledge graph making them readily available for use in downstream applications.

Preface

All work presented in this thesis is original work of the author, performed under the supervision of Prof. Laks V.S. Lakshmanan and Prof. Raymond Ng. The author of the thesis was the main author of this work and was also responsible for the implementation and analysis of this work. Dr. Michael Simpson, Prof. Raymond Ng, and Prof. Lakshmanan were involved in the project idea development, discussions and contributed to manuscript edits. Baoxing Huai, Zhefeng Wang, and Lanjun Wang, Huawei, were involved in the project discussions. Professor Giuseppe Carenini provided feedback on the final write-up of the thesis.

Table of Contents

Abstract iii

Lay Summary iv

Preface v

Table of Contents vi

List of Tables viii

List of Figures ix

Acknowledgements xii

1 Introduction 1

2 Related Work 5
2.1 Knowledge Graphs 5
2.2 Knowledge Graph Domains 5
2.3 Knowledge Source 6
2.4 Knowledge Graph Construction Techniques 6
2.4.1 Knowledge Extraction from Semi-structured Sources . . . 6
2.4.2 Knowledge Extraction from Unstructured Sources 6

3 The Knowledge Extraction Pipeline 9
3.1 Inputs 10
3.2 Text Pre-processing & Cleaning 11
3.3 Linguistic Annotations 14
3.4 Sentence Length Filtering 14
3.5 Semantic Role Labeling 14
3.6 Financial Predicate Dictionary Filtering 16
3.7 Temporal Argument Parsing 17
3.8 Appositions and Coordinating Conjunctions 17

3.9 Argument Validation 19
3.10 Pattern Extraction 22
3.11 Argument Minimization 25
3.12 Fact Scoring 26

4 Evaluation 28
4.1 Extraction Statistics 28
4.2 Precision 33
4.3 Knowledge Graph Statistics 33
4.4 Predicate and Patterns Distributions 34
4.5 Ablation Studies 49
4.6 Knowledge Graph Querying and Subgraph Visualization 54

5 Discussion and Future Work 59

6 Conclusions 63

Bibliography 64

List of Tables

3.1 Pattern types, instances and output extractions 24

4.1 Statistics and results of running our extraction pipeline vs the knowledge base of monetary transactions in [3] 31
4.2 Comparison between the functionalities of our pipeline vs [3]. . . 31
4.3 Financial lexicon hits 32
4.4 Knowledge graph statistics 33
4.5 Frames and predicates 37
4.6 The top ten most common pattern extracted facts 39
4.7 Sample relations extracted via patterns 49
4.8 Ablation studies 53

List of Figures

3.1 The financial knowledge extraction pipeline 10
3.2 FrameNet example 10
3.3 An example sentence for the roleset id *acquire.01*. The sentence is annotated with its roles as color coded below the sentence. 11
3.4 Sample entries from the Financial Domain Lexicon 12
3.5 Example of noisy text spans in news article leads. The colors highlight the noise spans boundaries. 13
3.6 Semantic roles for the arguments of the predicate acquire.01 in two different sentence. While the syntactic subject and object are different in both sentences, the argument roles are the same. 15
3.7 Entry for *acquire.01* in the financial predicate dictionary; "r" is *required* while "o" is *optional*. 17
3.8 The dependency parse tree for the sentence: *Dubai-based port operator DP World also announced plans to transport cargo using a functioning system*. the relation isA(DP World, Dubai-based port operator) is extracted by following the *APPO* dependency relation between *operator* and *World* 18
3.9 The dependency parse tree for the sentence: *HFF , HFF Real Estate Limited , HFF Securities L.P. and HFF Securities Limited are owned by HFF Inc.* Processing coordinating conjunctions produces four relations with simpler arguments rather than one big argument *HFF , HFF Real Estate Limited , HFF Securities L.P. and HFF Securities Limited.* 19
3.10 Example of a predicate-argument structure that does not pass the financial predicate filtering step. For the predicate *increase.01* as shown in its dictionary entry below the extraction, *A1* is required. . 20
3.11 An example of a predicate-argument structure that does not pass the financial predicate filtering step. For the predicate *sink.01* as shown in its dictionary entry below the extraction, *A4* must be one of the named entity types *MONEY*, *QUANTITY*, *CARDINAL* or *PERCENT* 21

3.12 An example of a predicate-argument structure that does not pass the financial predicate filtering step. For the predicate *increase.01* as shown in its dictionary entry below the extraction, since only one argument is required, i.e, *A1*, at least one more optional arguments must be present. 22
3.13 An example of an apposition extraction with and without minimization 26
3.14 An example of an extraction in which certain sub sequences of the pattern instance were found in the financial lexicon marking all of its tokens as stable. 26

4.1 Predicate distribution 35
4.2 Frames distribution 38
4.3 Pattern distribution 39
4.4 Top 20 relational nouns 40
4.5 Sample facts from the frame *Change_position_on_scale* 41
4.6 Sample facts from the semantic frame *Getting* 41
4.7 Sample facts from the semantic frame *Commerce_buy* 42
4.8 Sample facts from the frame *Make_agreement_on_action* 42
4.9 Sample facts from the semantic frame *Appointing* 42
4.10 Sample facts from the semantic frame *Commerce_sell* 43
4.11 Sample facts from the frame *Cause_change_position_on_scale* . . 43
4.12 Sample facts from the semantic frame *Intentially_create* 44
4.13 Sample facts from the semantic frame *Funding* 44
4.14 Sample facts from the semantic frame *Process_end* 44
4.15 Sample facts from the semantic frame *Supply* 45
4.16 Sample facts from the semantic frame *Activity_start* 45
4.17 Sample facts from the semantic frame *Earnings_and_losses* 45
4.18 Sample facts from the semantic frame *Using_resource* 46
4.19 Sample facts from the semantic frame *Commerce_pay* 46
4.20 Sample facts from the frame *Cause_change_of_strength* 46
4.21 Sample facts from the semantic frame *economic_value* 47
4.22 Sample facts from the semantic frame *Legal_action* 47
4.23 US-China trade. The relation highlighted shows that the *United States* exported *$15 billion dollar of aircraft* to *China* on *04/04/2016*. Section 4.4 discusses how we break down n-ary relations to create the knowledge graph. 55
4.24 Subgraph representing acquisitions by German drugmakers. The relation highlighted shows that the *Merck* acquired *Cubist Pharmaceuticals Inc.* on *04/04/2015*. 56

4.25 Companies suing on patent grounds. 57
4.26 Companies suing on patent grounds. 58

5.1 Co-reference resolution errors. 60

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Professor Laks V.S. Lakshmanan, for his kindness and patience and for his guidance and mentorship since I started my journey at UBC and throughout the course of this work.

I would like to thank my co-supervisor, Professor Raymond Ng and our collaborator Dr. Michael Simpson, for their valuable advice and support which were instrumental in completing this work.

I would like to thank Professor Giuseppe Carenini for providing feedback on the thesis which greatly helped in ensuring its consistency and clarity.

I would also like to thank the members of the Data Management and Mining Laboratory and the Social Networks Reading Group for the stimulating discussions and insightful comments.

And to my family, thank you for all your love and support.

This research was supported by a grant from Huawei Technologies Co. and UBC's Data Science Institute.

Chapter 1

Introduction

Knowledge graphs(KG) have lately emerged as a de facto standard for knowledge representation in the Semantic Web, whereby knowledge is expressed as a collection of "facts", represented in the form (subject,predicate,object) triples, where subject and object are entities and predicate is a relation between those entities. This collection can be conveniently stored, queried, and maintained as a graph, with the entities modeled as vertices and relations as links or directed edges.

Over the years, a number of *cross domain* knowledge graphs were created such as DBpedia [18], YAGO [29], Freebase [4], Wikidata [30] and NELL [22] covering millions of real world entities and thousands of relations across different domains such *people*, *organizations* and *geography*. These knowledge graphs were either manually curated (e.g Freebase and Wikidata), automatically created from semi-structured textual sources such as Wikipedia infoboxes and categorization information (e.g DBpedia and YAGO) or unstructured text on the web (e.g NELL). Despite their large scale, they were limited with a predefined ontology of entity types and relations. The need for knowledge graphs that are targeted towards particular domains and cover domain specific entities and relations remained strong. As a result, a number of *domain targeted* knowledge graphs were created using knowledge extraction pipelines that are tailored to the particular domain. These efforts include CPIE [31] which extracted relations between 3 fixed types of biomedical entities, Aristo Tuple knowledge base [21] which developed a high precision knowledge extraction pipeline targeted towards elementary science knowledge.

The financial domain was no exception and the interest in having structured representations for financial and business entities and how these entities are related to each other was ever-increasing. This vision led to Crunchbase [1] knowledge base that covered hundreds of thousands of business entities including companies and investors but only a few types of business transactions such as acquisitions and funding rounds. This limitation coupled with the fact that the data is provided through partnerships with companies and data experts limits both its utility and scalability. The financial domain also covers markets, stocks, currencies, commodities, funds, economies and government. Having structured representations of

[1] https://www.crunchbase.com/

these entities and transactions stored in a knowledge graph could help us answer interesting and complex queries such as

- Company acquisitions by German drugmakers.
- US-China trade in terms of exports.
- Companies suing each other on patent grounds.
- The longest span of increase/decrease of a company's stock price.
- The most/least profitable company acquired by a certain company based on net revenues.

The answers to these queries lie in the hundreds of news articles that are published every day. Manually extracting information from these sources is not only unscalable but also infeasible. [3] attempted to address some of the limitations of Crunchbase by developing a pipeline to populate a knowledge base *semi-automatically* with structured representations in the form of quintuples *(subject, predicate, object, monetary value, date)* extracted from a news corpus. It also trained a supervised model to rank different representations of the same economic event according to a confidence score. However, this pipeline only extracted 496 quintuples covering 316 economic events that fall into one of the two categories: events that increment the value of agent's resources (e.g. acquire, collect) or vice versa (e.g. pay, sell) with only 34% precision and 20% recall.

Our goal is to construct a *high precision* knowledge graph from financial information sources by extracting domain relevant (subject, predicate, object) triples. Our emphasis on achieving high precision is particularly important in Question Answering tasks where returning no results is more acceptable than returning incorrect results. Traditional approaches to knowledge graph construction from textual sources rely on pre-specified ontology of relations and large amounts of human annotated training data to learn extraction models for each relation which limits their scalability and their applicability to new relation types. *Open Information Extraction* (OpenIE) [2], aims to overcome these limitations by extracting *all* semantic relational tuples in raw surface form in a single pass over the corpus with little or no human supervision. Closely related to OpenIE is *Semantic Role Labeling* which aims at detecting argument structures associated with verb predicates. SRL, however, has the additional advantage that it labels the argument structures by their semantic role overcoming situations where the verb tense and conjugation change the role of the argument in the sentence. Consider the sentence: *Whole Foods was acquired by Amazon in 2017 for $13.7 Billion*.

- OpenIE: *(Whole Foods; was acquired; by Amazon; in 2017; for $13.7 Billion)*
- SRL: *acquire.01(agent: Amazon, thing acquired: Whole Foods, price paid: $13.7 Billion, Temporal argument: 2017)*

While OpenIE extraction is accurate, it's not useful for answering queries since it's not clear which entity acquired the other entity, or which argument is the price or the date. SRL, on the other hand, not only identifies the correct *sense* of the predicate *acquire.01* but also identifies the role of each argument. Not only this is helpful in question answering, it makes it possible to add structural and semantic constraints on the entity types to improve the precision.

We develop a *high precision* knowledge extraction pipeline tailored for the financial domain. The input to the pipeline is a financial news corpus and a financial lexicon that is used to minimize arguments that are considered *overly specific*. The pipeline cleans and removes noisy text spans that are commonly found in news articles. Then it extracts n-ary relational tuples via SRL and filters noisy predicate-argument structures via a dictionary of semantically and structurally constrained *sense-disambiguated* financial predicates. This financial dictionary filtering significantly ensures high precision and financially relevant extractions. Our pipeline produces more high precision extraction from (1) noun mediated relations *typed patterns* that operate over the part-of-speech tags and (2) implicit extractions from appositions (3) processing arguments with coordinating conjunctions. For maximizing the utility of the extractions for downstream applications, we minimize overly-specific arguments by utilizing the Financial Times Financial Lexicon. We score the extractions to reflect our confidence in them via supervised binary classifier. We perform a *lossless* decomposition of the n-ary relations to construct the knowledge graph.

Compared with [3], the most closely related work, our pipeline extracts over 342,000 n-ary facts and covers more types of financial predicates (e.g predicates describing change of position on a scale commonly used when reporting stocks performance such as rise, plummet, and predicates describing the action of granting or denying permissions such as approve, or sanction) – a total of 87 as opposed to 50 in [3]. Furthermore, our pipeline produces high precision extractions, specifically 78% at the top-100 extractions, as opposed to 34% of the pipeline from [3].

Our main contributions are:

- Developing a high precision knowledge extraction pipeline tailored to the financial news domain by combining the semantic role labeling and pattern based information extraction to effectively extract domain targeted noun/verb-

mediated relations. The ∼380,000 triples we extracted are stored in a KG which can be readily queried.

- Formulating noisy text cleaning as a sequence labeling problem and building a CRF model to remove noisy text spans which help improve the precision of the pipeline.
- Building a dictionary of structurally and semantically constrained sense-disambiguated financial predicates to filter out noisy SRL extractions and produce high precision extractions.
- Conducting ablation studies to examine the effect of the different modules of the pipeline on a number of metrics.

Chapter 2

Related Work

2.1 Knowledge Graphs

Towards the goal of the Semantic Web of having a *Web* of *Data* where semantic meaning is machine readable, knowledge graphs have emerged as a standard for knowledge representation. Over the years, a number of knowledge graphs have been created including Google Knowledge Graph [28], Google Knowledge Vault [7], DBpedia [18], YAGO [29], Freebase [4], Wikidata [30] and NELL [22]. We categorize them according to (a) *Domains of coverage* (b) *Knowledge source* (c) *KG construction techniques.*

2.2 Knowledge Graph Domains

Cross domain knowledge graphs such as Google Knowledge Graph, Knowledge Vault, YAGO, Freebase, Wikidata and NELL contain *encyclopedic* knowledge covering millions of real world entities (e.g people, organizations and geography) and thousands of relations. Several efforts to create *domain targeted* knowledge graphs have followed. HAVAS [2] launched an initiative [9] to create a knowledge graph that aggregates information about technology startups for analysis by media business experts. Aristo Tuple knowledge base [21] contains 294,000 (subject, predicate, object) tuples extracted using a high precision extraction pipeline targeted towards elementary science and guided by a domain vocabulary of 4th Grade Science texts. CPIE [31] extracted structured relational tuples between 3 biomedical entity types: gene, chemical, and disease. In the business and financial domain, Crunchbase [3] provides a knowledge base covering over 100,000 companies, investors, acquisitions and funding rounds. Towards building a knowledge base of monetary transactions, [3] extracted structured representations covering 316 economic events that fall into one of the two categories: events that increment the value of agent's resources (e.g. acquire, collect) or vice versa (e.g. pay, sell).

[2] https://havasmedia.com/

[3] https://www.crunchbase.com/

2.3 Knowledge Source

Freebase and Wikidata were created by a community of users who collaborated to add structured data. Similarly, the data in Crunchbase comes directly through partnerships with companies, a community of contributors and data experts. DBpedia and YAGO were automatically created from semi-structures sources (e.g. Wikipedia infoboxes, categorization information and external links). Knowledge Vault extracted triples from unstructured text documents in addition to HTML trees, HTML tables and human annotated pages. NELL started in Janurary 2010 with the task of populating a knowledge base with beliefs extracted from web pages from the ClueWeb2009 text corpus and Google search API. CPIE extracted the relational tuples from from PubMed paper abstracts and Aristo Tuple knowledge base extracted knowledge from the web using search templates for sentence selection parameterized by domain relevant entity types and instantiated by domain vocabulary.

2.4 Knowledge Graph Construction Techniques

2.4.1 Knowledge Extraction from Semi-structured Sources

YAGO extracts facts about (1) *Wikipedia entities* from infoboxes through manually created mappings from infoboxes attributes to YAGO relations (2) *entity types* from Wikipedia leaf categories (3) *Temporal information about the fact* from infoboxes using regular expressions. Similarly, DBpedia relies on a community curated ontology with mappings to Wikipedia information structures. It parses Wikipedia pages source code into Abstract Syntax trees that are passed to multiple high quality extractors for infoboxes, extractors for single features such as geographic coordinates, statistical extractors that employ statistical measures of page links or word counts to aggregate data from all Wikipedia pages. Knowledge vault extracts information from HTML tables by matching the predicates in columns to Freebase using standard schema matching techniques.

2.4.2 Knowledge Extraction from Unstructured Sources

Knowledge extraction from unstructured text involves (1) identifying the named entities in the text and their types, (2) resolving co-references of the same entity across documents, (3) defining the domain or ontology of semantic relations and entity types and (4) extracting the relations between the entities.

Knowledge vault uses a pre-defined ontology for predicates and entity types. It learns a logistic regression model for each predicate using distant supervision

which starts with a seed set of pairs of entities, from a distant knowledge base, that are related by the predicate in question and finds sentences where these pairs are mentioned and extracts features for the predicate from these sentences.

NELL learns to extract knowledge about semantic categories of entities and relations that exist between them by learning patterns that extract knowledge from web corpora and semi-structured data such as tables. It also learns Horn clauses to infer new relations from those that were already learned. The input to NELL is a predefined ontology of categories and relations and a few seed examples that are used for semi-supervised bootstrap learning of these relations. NELL integrates candidate facts from a pattern extractor for free text, a semi-structured extractor for lists and tables, a morphological classifier for noun phrases, and a first-order relational learner. NELL promotes candidate facts that have high confidence from a single source or lower-confidence candidate facts if they were proposed by multiple sources. These bootstrapping approaches used by Knowledge Vault and NELL reduce the human labour required to label train data while taking advantage of the huge amount of unlabeled data available.

While manually defining the ontology leads to high precision extractions, it requires domain experts and, in an open domain, it is not feasible to define a complete ontology. This limits the scalability of these approaches and their applicability to new relation types. OpenIE, an information extraction paradigm [2], aims to overcome these limitations by extracting *all* relational phrases in raw surface form with little or no human supervision. However, the verb tense and conjugation can change the role of an argument in the relation. SRL helps disambiguate the relations between arguments and their predicates by identifying semantic frames within the sentence and the semantic roles of the arguments.

The KB built in [3] used a pipeline that consisted of: (1) a grammar for monetary value recognition, (2) SRL for economic event identification, (3) entity recognition via DBpedia, Crunchbase and Freebase, and (4) date extraction via a temporal tagger. It extracted structured representations of economic events in the form of *(subject, predicate, object, monetary value, date)* quintuples from the New York Times Annotated Corpus (NYTC)[4]. It ranked all representations of an economic event according to confidence scores learnt using a supervised model. The domain was defined by a list of financial predicates using a semi-supervised method that starts with a set of seed predicates and expands them using WordNet [20]. The pipeline only extracts 496 quintuples from 316 economic events over just 2 categories of events, with a precision of 34%. Our work has the following key differences with prior work. We deal with the challenging task of identifying and removing noisy text spans in news articles. Our pipeline combines SRL and pattern

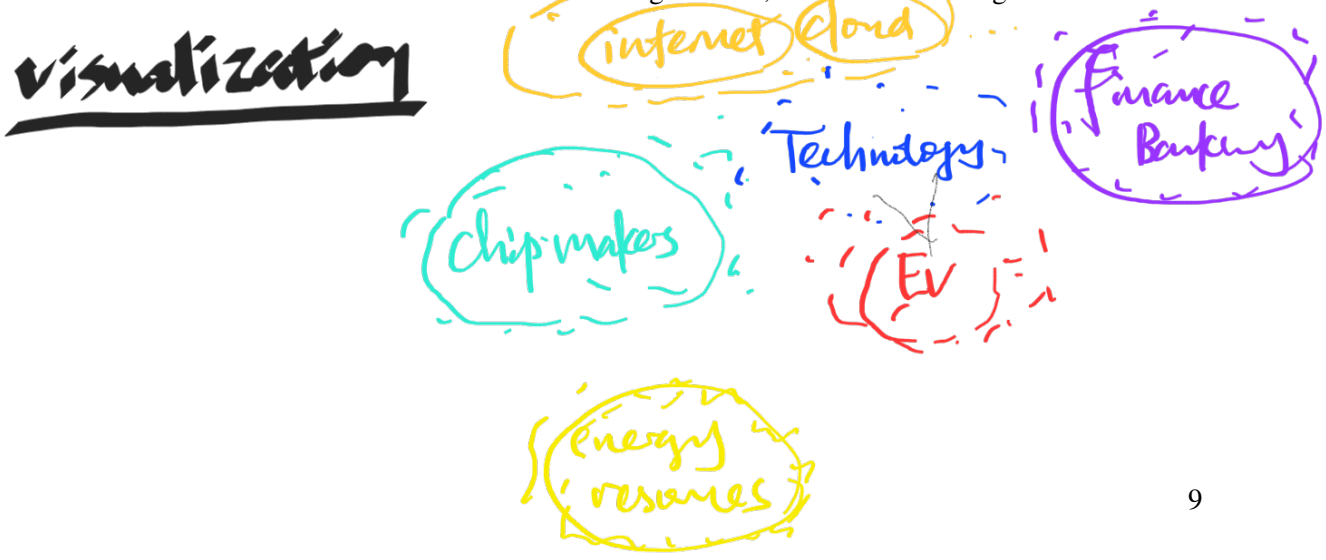
[4]https://catalog.ldc.upenn.edu/LDC2008T19

based IE in addition to producing implicit extractions from appositions. We build a dictionary of 87 semantically and structurally constrained financial predicates covering broader financial transactions and improving precision. We resolve coordinating conjunctions and do a dictionary-guided minimization to prune overly specific arguments. In all, our approach leads to a large knowledge graph with high precision. All these components work together to produce over 342,000 facts with 78% precision@100.

Chapter 3

The Knowledge Extraction Pipeline

In this chapter, we describe the pipeline for extracting structured representations in the form of n-ary relations from the financial news articles. As shown in Figure 3.1 The pipeline operates on sentence level and consists of multiple stages, each playing a role in improving the precision, the total number of extracted facts and/or the utility of the KG. The pipeline proceeds by first cleaning the news articles using a combination of regular expressions and a supervised learning model to remove noise. References of the same entities are then resolved and named entities are identified and sentences that are not within a range of length are filtered. The annotated text is then fed to a semantic role labeling system which identifies predicates, their senses, their arguments and the corresponding semantic roles. The financial predicates dictionary then filters noisy predicate argument structures that do not meet predicate-argument constraints. We produce additional extractions via high-precision typed patterns that are tailored to the financial domain, and by resolving appositions. We maximize the utility of the extractions by minimizing overly specific arguments by processing coordinating conjunctions and financial lexicon guided minimization and parsing temporal arguments into a standard date format. The extracted predicate argument structures are then scored to reflect our confidence in them. In the following sections, we describe the stages in more detail.



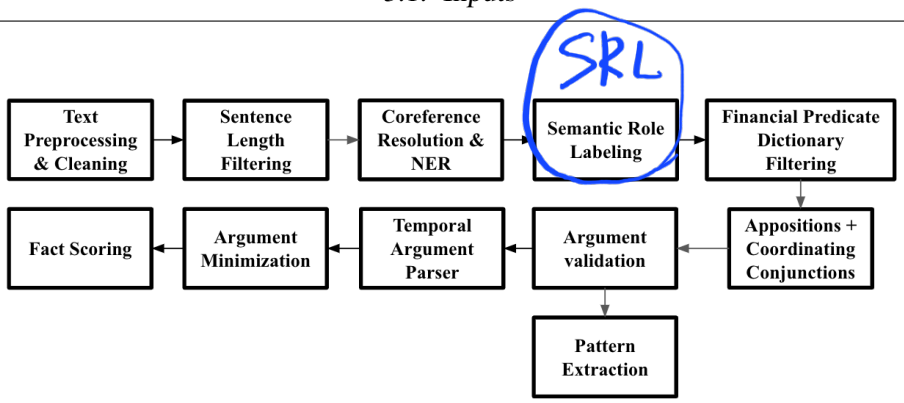


Figure 3.1: The financial knowledge extraction pipeline

3.1 Inputs

The input to our high precision knowledge extraction pipeline consists of

1. **Financial news corpus**: Kaggle US Financial news dataset. This dataset contains $\sim$ 306k news articles collected from Bloomberg.com, CNBC.com, reuters.com, wsj.com, and fortune.com between January and May 2018;

In November 2000 , as part of a $600 million agreement , Libya allegedly ACQUIRED the first shipment of a total of 50 North Korean Nodong ballistic missiles , including launch capabilities .

Explanation Manner Recipient Theme Time

Figure 3.2: Example of a sentence that evokes the frame *Getting*. The lexical unit that evokes this frame is the verb acquire. The sentence is annotated with the frame elements as color coded below the sentence.

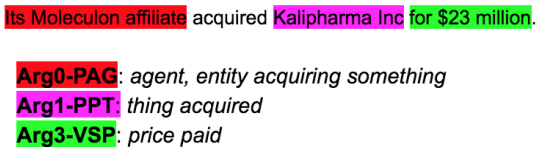


Figure 3.3: An example sentence for the roleset id *acquire.01*. The sentence is annotated with its roles as color coded below the sentence.

Listing 3.1: Sample entries in the financial predicates dictionary

```
{"acquire.01": {"A0":["r","ORG"], "A1":["r","ORG
   "], "A2":["o","ORG"], "A3":["o","MONEY"]},
"secure.01": {"A0":["r","ORG"], "A1":["r","MONEY
   "]},
"buy.01":{"A0":["r"], "A1":["r"], "A3":["o","MONEY
   "]},
"found.01": {"A0":["r","PERSON"], "A1":["r","ORG
   "]},
"finance.01": {"A0":["r","ORG|GPE"], "A1":["r"]},
"fund.01": {"A0":["r","ORG|GPE|PERSON"], "A1":["r
   "]},
"invest.01": {"A0":["r","ORG|GPE|PERSON"], "A1":["
   r", "MONEY"],"A2":["r"]},
"locate.01": {"A1":["r","ORG"],"AM-LOC":["r"]}}
```

2. **Financial domain lexicon**: we use the Financial Times Lexicon [5] which includes thousands of words and phrases relevant to the financial domain selected by Financial Times editors. We use this lexicon to identify and minimize overly specific arguments as described in the argument minimization stage in Section 3.11.

3.2 Text Pre-processing & Cleaning

The first step in the pipeline is data cleaning.We start with standard NLP cleaning by removing brackets, parenthesis, quotes and other punctuation marks. A more challenging yet important cleaning task that is unique to our problem is to identify

[5] https://markets.ft.com/glossary/searchLetter.asp?letter=A

- amortisation
- capital expenditure
- Bank of England
- bonds
- budget deficit
- capital ratio
- carbon trading
- commercial bank
- recession
- corporate bond
- environmental fund
- credit squeeze
- derivatives
- inventory cycle
- Dow Jones
- earnings per share
- EPS
- The Fed
- externality
- EBITDA
- Economic and Monetary Union
- EMU
- free market
- price/earnings ratio
- futures contract
- gross domestic product
- government bonds
- investment bank
- like-for-like sales
- market study
- Moody's
- operating profit
- pension fund
- pre-tax profit
- preferred share
- procurement
- purchasing power parity PPP
- rights issue
- share price
- shareholder
- taxpayer dollars
- tier one capital
- tier two capital
- Treasury
- underwriting power
- universal bank
- Wall Street

Figure 3.4: Sample entries from the Financial Domain Lexicon

and remove noisy text spans in the text. Most news articles include information about the publication date, the time the article was last updated, reporters' names, reading time, image captions, and other tags. This information is usually embedded within the article lead and is not separated from the content by punctuation marks as shown in Figure 3.5. We regard these text spans as noise since they only distort the text and negatively impact the precision of later stages. Variations in which this type of noise can appear in text limits the feasibility of using regular expressions to capture these noisy text spans. Additionally, this type of noise can appear anywhere in the article and is not limited to the article lead.

We address this problem using supervised learning. We cast the task of identifying noisy text spans as sequence labeling where sequences of tokens are assigned sequences of labels. Following the IOB tagging format, a token is assigned (I) if it's inside a noisy text span, (B) if it's the beginning of the span or (O) if it's outside the span. We build a Conditional Random Field (CRF) [17] model and use the CRFSuite implementation [24] of linear chain CRF to label sequences of tokens. We extract a number of features for each token that combine information from the surrounding tokens and their part-of-speech tags. CRFsuite learns associations between these features and the corresponding labels. We use Stanford CoreNLP library [19] for tokenization, sentence segmentation and part-of-speech tagging. We build a training dataset of 779 sentences with 37% noise. The model is then used to label tokens and noisy text spans are then removed. While more sophisticated neural sequence labeling approaches that incorporate semantic meaning using word embeddings could be used, we found that the CRF model, while simple, proved to be powerful in detecting the noisy text spans and ultimately improving the precision of the pipeline as will be shown in the ablation studies.

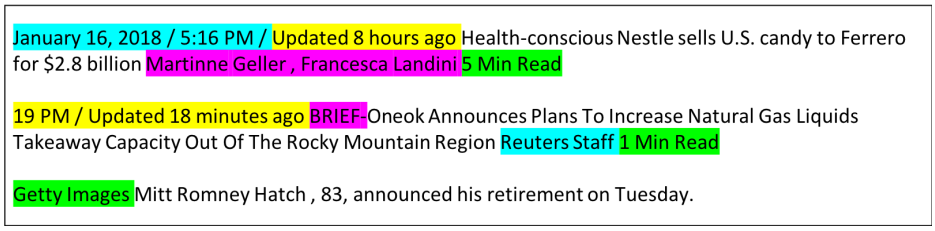


Figure 3.5: Example of noisy text spans in news article leads. The colors highlight the noise spans boundaries.

3.3 Linguistic Annotations

Once the text is cleaned, we then resolve references of the same entity using the co-reference resolution system integrated in SpaCy NLP library [12] that is based on a neural mention-ranking model optimized using deep reinforcement learning [6]. We identify named entities using a state-of-the-art AllenNLP fine-grained named entity recognition (NER) system that covers 16 entity types based on Bidirectional LSTM neural architecture with CRF layer and ELMo embeddings [26]

3.4 Sentence Length Filtering

We perform a sentence length filtering step in order to exclude sentences that are unlikely to contain meaningful facts. The following are examples of such sentences:

- ”The risks and uncertainties which forward-looking statements are subject to include, but are not limited to: changes in general economic, business and political conditions, including changes in the financial markets; weakness or adverse changes in the level of real estate activity, which may be caused by, among other things, ...”
- “Reporting by Eric Walsh .”

Based on empirical observations, we exclude sentences that are longer than 45 tokens and shorter than 5 tokens in length.

3.5 Semantic Role Labeling

Once the text is cleaned and annotated with resolved co-references and named entities, we then turn our attention to extracting semantic relationships between these entities. As described in Section 1, while the *OpenIE* paradigm is capable of extracting semantic relational tuples with little to no human supervision in a single pass over the corpus, the verb tense and conjugation can change the role of an argument in the relation (i.e whether the argument is an *agent* which carries out the predicate action or a *theme* which receives the predicate action). Frame Semantics theory helps disambiguate the relations between arguments and their predicates by identifying semantic frames within the sentence and the thematic roles of the arguments, i.e., Frame Elements, based on the syntactic structure and the lexical units that evoke this frame as shown in Figure 3.2 (e.g., *acquire*).

In Figure 3.6, while the tense and conjugation are different in the two sentences, the role of *Amazon* is the *agent* or *the entity acquiring something* in both sentences although it is a syntactic subject in one but a syntactic object in the other. Similarly, *Whole Foods* is the *theme* or the *thing acquired*. Semantic role labeling also identifies that *$13.7 billion* was *the price paid* and *2017* is the temporal argument using Propbank annotations.

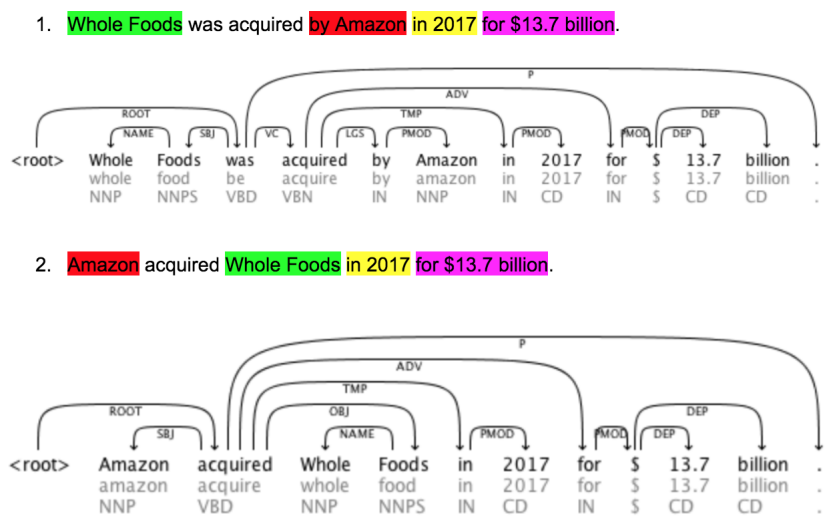


Figure 3.6: Semantic roles for the arguments of the predicate acquire.01 in two different sentence. While the syntactic subject and object are different in both sentences, the argument roles are the same.

Correctly identifying the sense of the predicate and thematic roles of its arguments helps us impose *structural* and *semantic* restrictions to improve the precision of the extracted relations. For the predicate *acquire*, we know that it must have at least two arguments, the *entity acquiring something* and the *thing acquired*, we know that both of these arguments, in the financial domain, have to be organizations, *the price paid* has to be money. Figure 3.7 shows the entry of the predicate *acquire* in the financial predicates dictionary. *Information about the predicate*

sense and thematic roles of its arguments in the relation are not captured by traditional OpenIE extractors which make them less useful particularly in domain specific information extraction as it limits the ability to impose domain specific structural and semantic constraints on the arguments which help improve the precision.

We use the LUND Propbank Semantic Role Labeling system (LUND-SRL) [13], the winner of CoNLL 2008 shared task achieving $\sim$ 85% F1-score which is comparable to more recent neural models such as [11], for extracting and labeling predicate sense-argument structures. LUND-SRL carries out joint dependency-syntactic and semantic analysis consisting of a pipeline of predicate disambiguation, argument identification, and argument labeling classifiers followed by linguistically motivated global constraints to filter the candidates generated by the pipeline then a global reranker that scores predicate–argument structures in the filtered candidate list. The output from LUND-SRL include the predicted lemma, predicted part-of-speech tags, predicted dependency relations, predicate sense, and argument labels.

3.6 Financial Predicate Dictionary Filtering

We filter out domain irrelevant predicate-argument structures using a dictionary of financial predicates. This dictionary lists the sense-disambiguated predicates along with structural contraints, i.e., required vs optional arguments, and semantic constraints, i.e., the possible entity types (e.g., *ORG*, *MONEY*). We construct the dictionary by automatically extracting sense-disambiguated predicates from the corpus and manually selecting the highest frequency ones that are relevant to the financial domain. We expand this set using the FrameNet [1] lexical resource. This yields 87 financial predicates. For each of these sense-disambiguated predicates, we determine the required arguments and potential entity types using Propbank semantic roles annotations [16]. Fig. 3.7 shows the entry for *acquire.01* in the dictionary. It is important to note that this dictionary is different from the financial lexicon we described earlier as an input to the pipeline. As will be described below, the lexicon will guide the minimization of arguments that are considered overly specific, whereas this dictionary filters out predicate argument structures that are either not financially relevant or are not in compliance with the semantic and structural constraints. We filter out predicate-argument structures that contain modal arguments (e.g., *Google could have acquired Facebook*) or negated arguments (e.g., *Google did not acquire Facebook*) since these structures are unlikely to represent facts. Furthermore, we only include structures where the predicate is in past tense (e.g., *Google acquired Youtube*). We also filter out predicate-argument

structures with adverbial arguments (AM-ADV) representing adverbs of negation such as *"hardly", "never", "almost"* since they do not represent *positive* facts (e.g., *Yahoo almost acquired Facebook*).

```
"acquire.01": { "agent, entity acquiring something": ["r", "ORG"],
                "thing acquired": ["r", "ORG"],
                "seller": ["o", "ORG"],
                "price paid": ["o", "MONEY"] }
```

Figure 3.7: Entry for *acquire.01* in the financial predicate dictionary; "r" is *required* while "o" is *optional*.

3.7 Temporal Argument Parsing

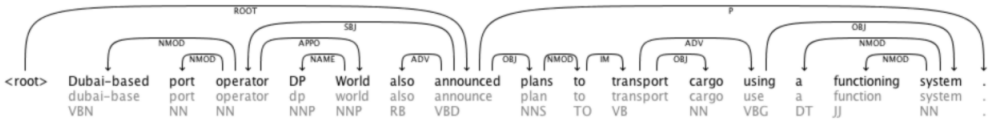
Many of of the SRL extracted relations contain temporal arguments *AM-TMP* such as *today*, *last year*, *3 months ago*. We pass these arguments to a date parser library [6] that performs relative parsing of localized dates in numerous formats to produce a standard date format. We use the publication date as the relative base. While not all arguments could be parsed to a standard format, we perform this step to maximize the utility of the extracted facts in downstream tasks.

3.8 Appositions and Coordinating Conjunctions

We produce implicit extractions from appositions in arguments by following the dependency relation *APPO* as shown in Figure 3.8 and determining the argument boundaries as described in section 3.5.

[6] https://pypi.org/project/dateparser/

Dubai-based port operator DP World also announced plans to transport cargo using a functioning system .



Appositions: isA(DP World, Dubai-based port operator)

Figure 3.8: The dependency parse tree for the sentence: *Dubai-based port operator DP World also announced plans to transport cargo using a functioning system*. the relation isA(DP World, Dubai-based port operator) is extracted by following the *APPO* dependency relation between *operator* and *World*

We process coordinating conjunctions which join similar syntactic units (i.e *conjoints*) into larger groups by means of coordinating conjunctions such as *and*, *or*. Consider the example in Figure 3.9, *HFF , HFF Real Estate Limited , HFF Securities L.P. and HFF Securities Limited* is an overly specific argument with four conjoints *HFF* , *HFF Real Estate Limited* , *HFF Securities L.P.* and *HFF Securities Limited* joined by the coordinating conjunction *and*. Processing coordinating conjunctions (i.e extracting conjoints to simplify the relation) aims at both simplifying the overly specific arguments and increasing the recall of the pipeline by producing more facts. In Figure 3.9 processing coordinating conjunctions produces four simpler relations instead of one with an overly specific argument.

We find the coordinating conjunctions by the dependency relations *coord* and *conj*. We find the first conjoint head by following the dependency relation *conj* and the other conjoint heads by following the dependency relation *coord*. Proper processing of coordinating conjunctions is a challenging task especially when there's more than one coordinating conjunction in the argument. We limit the processing to only the first coordinating conjunction in the argument and we do not process coordinating conjunctions whose head is *between* (ex. *Costco Wholesale fell between 1.3 percent and 3.7 percent*).

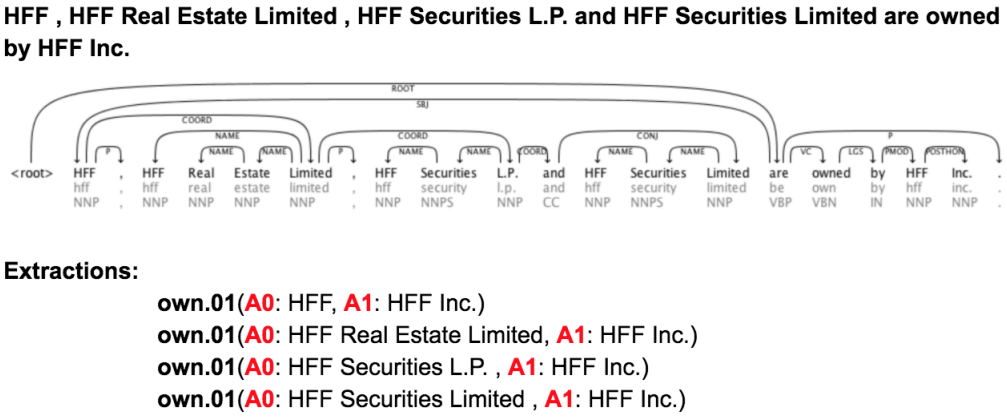


Figure 3.9: The dependency parse tree for the sentence: *HFF , HFF Real Estate Limited , HFF Securities L.P. and HFF Securities Limited are owned by HFF Inc.* Processing coordinating conjunctions produces four relations with simpler arguments rather than one big argument *HFF , HFF Real Estate Limited , HFF Securities L.P. and HFF Securities Limited.*

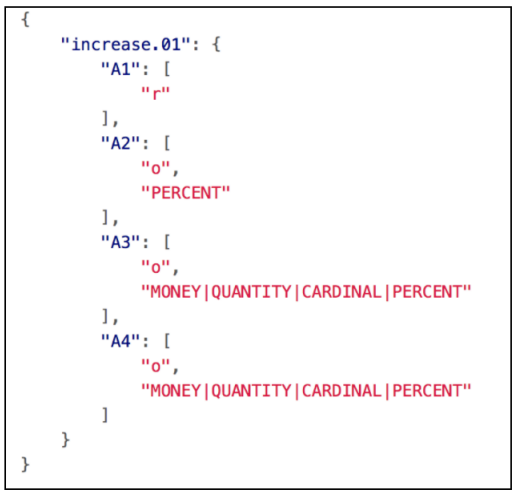
3.9 Argument Validation

Not only does this dictionary filtering step helps ensure that the facts are relevant to the financial domain, but also its semantic and structural constraints help eliminate predicate-argument structures that we incorrectly labeled by the SRL system. In Figure 3.6, arguments *A0* (*Amazon)*) and *A1* (*Whole Foods*) were correctly identified, both satisfying the entity type constraint *ORG*. Similarly, A3 (*$13.7 billion*) is *MONEY* and *AM-TMP* is *DATE*. Consequently, both predicate-argument structures pass the Financial Predicate Filter. Figure 3.10 shows a predicate-argument structure that fails to pass the financial predicate filtering step. The predicate *increase.01* requires argument *A1* which is represents the semantic role *thing increasing*. In this example, SRL misclassifies *Adjusted EBITDA of the rental segment* as *A0* which represents *causer of increase* not the *thing increasing*. As a result, this predicate argument structure does not pass the filtering step. This instance shows the role of the filtering step not only in improving the relevance of the extracted facts but also the precision of the overall pipeline. Figure 3.11 shows another example that does not pass the predicate filtering step since one of the extracted arguments does not satisfy the entity type constraints. Figure 3.12 shows another example where none

of the optional arguments specified in the dictionary were extracted so it fails to pass the filtering step.

Adjusted EBITDA of the rental segment increased by 19.1 % in the compared period .

A0: Adjusted EBITDA of the rental segment
A2: by 19.1 %



```
{
    "increase.01": {
        "A1": [
            "r"
        ],
        "A2": [
            "o",
            "PERCENT"
        ],
        "A3": [
            "o",
            "MONEY|QUANTITY|CARDINAL|PERCENT"
        ],
        "A4": [
            "o",
            "MONEY|QUANTITY|CARDINAL|PERCENT"
        ]
    }
}
```

Figure 3.10: Example of a predicate-argument structure that does not pass the financial predicate filtering step. For the predicate *increase.01* as shown in its dictionary entry below the extraction, *A1* is required.

In 2000 , Israeli-Palestinian relations sank to a low with the outbreak of the second Palestinian intifada .

A1: Israeli-Palestinian relations
A4: to a low


```
{
    "sink.01": {
        "A1": [
            "r"
        ],
        "A2": [
            "s",
            "PERCENT"
        ],
        "A3": [
            "s",
            "MONEY|QUANTITY|CARDINAL|PERCENT"
        ],
        "A4": [
            "s",
            "MONEY|QUANTITY|CARDINAL|PERCENT"
        ]
    }
}
```


Figure 3.11: An example of a predicate-argument structure that does not pass the financial predicate filtering step. For the predicate *sink.01* as shown in its dictionary entry below the extraction, *A4* must be one of the named entity types *MONEY*, *QUANTITY*, *CARDINAL* or *PERCENT*

The erosion of arms control agreements , deployment of additional weapons and tensions over military exercises have increased the risk of an inadvertent armed clash between Europe and Russia , according to the annual Munich Security Report .

A0: The erosion of arms control agreements
A0: deployment of additional weapons and tensions over military exercises
A1: the risk of an inadvertent armed clash between Europe and Russia


```
{
    "increase.01": {
        "A1": [
            "r"
        ],
        "A2": [
            "o",
            "PERCENT"
        ],
        "A3": [
            "o",
            "MONEY|QUANTITY|CARDINAL|PERCENT"
        ],
        "A4": [
            "o",
            "MONEY|QUANTITY|CARDINAL|PERCENT"
        ]
    }
}
```


Figure 3.12: An example of a predicate-argument structure that does not pass the financial predicate filtering step. For the predicate *increase.01* as shown in its dictionary entry below the extraction, since only one argument is required, i.e, *A1*, at least one more optional arguments must be present.

3.10 Pattern Extraction

In addition to the verb mediated relations extracted via semantic role labeling, we extract noun mediated relations via pattern based extractor. The patterns are similar to those in [25] part-of-speech and noun chunks extractor except that we add entity type constraints to the patterns. Specifically, we extract the patterns shown in table 3.1. The addition of the named entity type constraint, i.e *ORG* and *PER*, yields high precision extractions through patterns that are commonly found in financial news. Furthermore, it facilitates segmenting compound relational nouns that are not preceded by a demonym, i.e., words derived from the name of a place

to identify its residents or natives), e.g., *Canadian*, *North American*, etc.. Consider the phrase *"Apple Chief Executive Officer Tim Cook"*, The pattern extractor would extract the relation *(Tim Cook, is a Chief Executive Officer, of/from Apple)*. Knowing that *Tim Cook* is a named entity of type *PER* and *Apple* is a named entity of type *ORG*, the pattern extractor would match the phrase with the pattern type *ORG-PER compound relational Noun*

For the patterns that contain demonyms (i.e words derived from the name of a place to identify its residents or natives), ex. *Canadian*, *North American*, we use the demonym-location table from [25] that was primarily populated from the list of demonyms from Wikipedia. We create a string the replaces tokens with POS tags, named entity types or demonyms if present, then check if it matches one of the patterns in table 3.1. So the tokens in the phrase *"Apple Chief Executive Officer Tim Cook"* would be replaced by *"ORG NNP NN NN PERSON PERSON"* and matched against the pattern types in the table.

Pattern type	Pattern instance	Extraction
Demonym-ORG compound relational noun	Canadian multinational corporation Bombardier	(Bombardier, is a multinational corporation, from Canada)
Demonym-PER compound relational noun	Canadian Prime Minister Justin Trudeau	(Justin Trudeau, is a Prime Minister, of/from Canada)
Demonym-ORG possessive	Canada's multinational corporation Bombardier	(Bombardier, is a multinational corporation, from Canada)
Demonym-PER possessive	Canada's Prime Minister Justin Trudeau	(Justin Trudeau, is a Prime Minister, of/from Canada)
Demonym-ORG possessive appositive	Canada's multinational corporation, Bombardier	(Bombardier, is a multinational corporation, from Canada)
Demonym-PER possessive appositive	Canada's Prime Minister, Justin Trudeau	(Justin Trudeau, is a Prime Minister, of/from Canada)
ORG-PER compound relational Noun	Apple Chief Executive Officer Tim Cook	(Tim Cook, is a Chief Executive Officer, of/from Apple)
ORG-PER appositive	Apple Chief Executive Officer, Tim Cook	(Tim Cook, is a Chief Executive Officer, of/from Apple)
ORG-PER possessive appositive	Apple's Chief Executive Officer, Tim Cook	(Tim Cook, is a Chief Executive Officer, of/from Apple)
PER-ORG possessive appositive	Tim Cook, Apple's Chief Executive Officer	(Tim Cook, is a Chief Executive Officer, of/from Apple)
PER-ORG appositive	Tim Cook, Apple Chief Executive Officer	(Tim Cook, is a Chief Executive Officer, of/from Apple)

Table 3.1: Pattern types, instances and output extractions

3.11 Argument Minimization

We then turn our attention to minimizing *overly specific* extractions in order to maximize their utility in downstream tasks. In this step we identify and drop certain parts from the argument that are considered overly specific. Similar to [8], we begin by dropping tokens that are considered safe to drop. This includes determiners, possessives, adjectives modifying named entity PER except demonyms. We then perform dictionary minimization. We first find instances of the noun phrase pattern

$$[adverbial|adjective]^{+}\ Noun^{+}$$

then for every instance of this pattern, we mark the root and all its noun modifiers as stable, then we enumerate sub sequences of this instance where at least one noun is retained. Also if a noun is not retained then neither its modifiers. We then query these sub sequences against a dictionary, if it is found in the dictionary, then all its tokens are marked as stable. After querying all sub sequences, we drop tokens that are not marked as stable.

While [8]'s dictionary mode uses a dictionary of frequent subjects, relations and arguments found in a corpus, we use the Financial Times financial lexicon that consists of over 13,000 entries of nouns and phrases relevant to the financial domain. This ensures that we do not drop tokens that are meaningful and important in the financial context.

Figure 3.13 shows an example of an extraction where one of the arguments is overly specific (i.e *an advanced technology company*). The minimization module first drops the determiner *an*, then proceeds with enumerating sub sequences for the noun phrase pattern instance *advanced technology company* and querying its instances against the financial lexicon, and determines that *advanced* is not a stable sub constituent of the pattern. Figure 3.14 shows an extraction where subsequences of the noun phrase pattern instance *foreign direct investment* were found in the financial lexicon (i.e *direct investment* and *foreign direct investment*) and subsequently all of its tokens are marked as stable.

"Mikros Systems Corporation , an advanced technology company, today announced financial results for 2017."

Appositions without minimization:
isA(Mikros Systems Corporation, an advanced technology company)

Appositions with minimization:
isA(Mikros Systems Corporation, technology company)

Figure 3.13: An example of an apposition extraction with and without minimization

"U.S. foreign direct investment into Mexico has increased from $ 15 billion to more than $ 100 billion"

Pattern instance: *foreign direct investment*
Increase.01

A1: U.S. foreign direct investment into Mexico
A3: from $ 15 billion
A4: to more than $ 100 billion

Figure 3.14: An example of an extraction in which certain sub sequences of the pattern instance were found in the financial lexicon marking all of its tokens as stable.

3.12 Fact Scoring

We then proceed with scoring the predicate argument structures to reflect our confidence in them. We train a binary logistic regression classifier over a dataset of over 1400 SRL extracted predicate-argument structures that we manually labeled. Facts are considered valid if they are both *precise* and *concise*, i.e, explains only one proposition, in other words extractions should be compact with structures like conjunctions and oppositions properly resolved. By examining the extracted relations, we identified some features that are powerful predictors of the validity of the extraction. The features include the presence of a coordinating conjunction,

apposition, verb, unresolved temporal arguments, pronouns, determiners, bad characters, if the argument length is more than 10 tokens in length, the predicate and named entities in the argument. We classify each valid argument of the extracted fact and take the minimum of all arguments' scores as fact confidence score. We chose the minimum as our aggregate function in order to promote the most precise facts.

Chapter 4

Evaluation

We run our knowledge extraction pipeline on the US Financial News Articles [7] described in section 3.1. In the following sections, we report the extractions statistics, precision, knowledge graph statistics, distributions of predicates, semantic frames and patterns. Further, we compare the functionalities of our pipeline against [3] in Table 4.2. We also conduct ablation studies to show the effect of the different stages of the pipeline on a number of metrics and discuss querying the knowledge graph via Datalog and visualize the resulting subgraphs. We do not report the recall since it is not feasible to find the ground truth facts in the corpus.

We also compare some of the results with the work in [3] which extracts structured representations of economic events in the form of (subject, predicate, object, monetary value, data) from the New York Times Annotated Corpus (NYTC) [8], which contains over 1.8M news articles published by the New York Times between 1987 and 2007, using a natural language processing pipeline. This pipeline considers all possible representations of an event and uses a supervised learning approach to rank these representations by their confidence score. The code for this pipeline was not made publicly available so we were not able to run it on our corpus for accurate comparison. The significance of the statistics presented in this chapter will be discussed in Section 5.

4.1 Extraction Statistics

To demonstrate the effectiveness of the pipeline, we report a number of extraction statistics in table 4.1 resulting from the of processing 288,118 articles. These statistics are influenced by the different modules of the pipeline. A total of 342,181 facts were extracted by the SRL, pattern and appositions modules from 5.2% of of sentences with average arity of 2.27. On the other hand, the pipeline in [3] that consisted of grammmar for monetary value recognition, semantic role labeling for economic event identification via semantic role labeling, entity recognition via DBpedia, Crunchbase and Freebase, and date extraction via Stanford Tempo-

[7] https://www.kaggle.com/jeet2016/us-financial-news-articles
[8] https://catalog.ldc.upenn.edu/LDC2008T19

ral Tagger, extracted 496 quintuples from 18.2% of the sentences at 20% recall. Of the predicate-argument structures that were eliminated, 94.7% did not pass the financial predicate filtering step. This indicates that the financial dictionary filtering had the greatest impact on the recall of the pipeline. It also suggests that the corpus does not contain enough financially relevant information. We examine the effect of turning off the financial predicate filtering in the ablation studies in section 4.8. Another 3.69% of the relations were eliminated as a result of not satisfying the requirement that predicates should be in the past tense and are not negated nor contain modal arguments or adverbs of negation whereas 1.52% did not pass the predicate role validation step. This suggests that the semantic and structural constraints on the predicates do not play a major role in filtering candidate predicate argument structures, hence relaxing these constraints will not substantially increase the number of extractions. Similar to our approach, the work in [3] created a list of financial predicates using a semi-supervised method that starts with a set of seed predicates and expands them using WordNet. Their approach, however, was limited to only two types of economic events, i.e events that increment the value of agent's resources (e.g. acquire, collect) or vice versa (e.g. pay, sell). Our financial predicate dictionary, on the other hand, not only includes more types of economic events (e.g rise, fall) but also places structural and semantic constraints on the financial predicates which play a major role in the precision of the extractions.

More than half of facts were implicitly extracted via appositions. We examine turning off the appositions module on a number of metrics including the precision in the ablation study. The SRL module extracted over 161,000 predicate argument stuctures contributing to 47.83% of the facts, whereas $\sim$1.3% were noun mediated relations extracted via patterns. It's important to note that of the $\sim$11,000 pattern extracted facts, 4454 are distinct. The work in [3], however, only uses semantic role labeling for extracting economic events in addition to recognizing noun predicates that originate from verbs using the NomBank dataset. The minimization module responsible for minimizing overly specific arguments dropped over 427,000 tokens. Over half of these tokens were safe minimizations, i.e. determiners, possessives, adjectives modifying named entity PER except demonyms. The rest of the tokens were dropped by the dictionary minimization which queries sub sequences of adverbial or adjective phrases against the financial lexicon 3.4 and marks all its tokens as stable if found in the lexicon. Table 4.3 shows the top bigram lexicon hits. *quarterly dividend* bigram was queried 372 times, followed by *common stock* and *net income* and subsequently its tokens were marked as stable. The adjectives in these bigrams, i.e. *quarterly*, *net* and *common* are critical in financial context, hence dropping them would result in the loss of important information which is alleviated by querying against the financial lexicon. This emphasizes the importance of the financial lexicon in preserving tokens that are important in financial

context while minimizing overly specific arguments. We also discuss the effect of the minimization module on the average argument length in section 4.8.

	Our Pipeline	Monetary KB [3]
Corpus	US Financial News Articles	NYTC
Number of articles processed	288,118	1,800,000
Number of sentences processed	3,884,948	2,100,000
Number (%) of sentences with extracted facts	201,731 (5.2%)	383,000 (18.2%)
Number (%) of temporal arguments that were parsed to standard date format	51,505 (78.5%)	N/A
Number of pattern extracted facts	10,828	0
Number (%) of *distinct* pattern extracted facts	4,454 (1.3%)	0
Number (%) of SRL extracted facts	161,983 (47.33%)	496 (100%)
Number (%) of appositions extracted facts	175,744 (51.35%)	0
Total number of facts	342,181	496
Average argument length in tokens	3.68 tokens	N/A
Number of tokens dropped by minimization stage	427,337	N/A
Number (%) of tokens dropped by safe minimization	227,368 (53.2%)	N/A
Number (%) of tokens dropped by Dictionary minimization	109,971 (46.8%)	N/A
Precision@50	78%	N/A
Precision@100	78%	N/A

Precision@150	79.33%	N/A
Precision	N/A	34%
recall	N/A	20%
F1	N/A	25%

Table 4.1: Statistics and results of running our extraction pipeline vs the knowledge base of monetary transactions in [3]

	Our Pipeline	[3]
Cleaning	✓	✗
Co-reference Resolution	✓	✗
Named Entity Recognition	✓	monetary value recognition
Appositions	✓	✗
Coordinated Conjunctions	✓	✗
Financial Predicate Dictionary Filtering	87 predicates with semantic and structural constraints	50 predicates, two types of econ. events
Pattern Extraction	✓	✗
Argument Minimization	✓	✗
Fact Scoring	✓	✓

Table 4.2: Comparison between the functionalities of our pipeline vs [3].

Bigrams	**Number of hits**
quarterly dividend	372
common stock	354
net income	277
cash dividend	239
net sales	172
public offering	131
fiscal year	118
global leader	96
central bank	84
vice president	79
gross profit	74
net profit	68

annual report	62
registration statement	61
intangible assets	60
joint venture	58
initial public offering	54
real estate	52
common share	50
chief executive	49
gross margin	49
outstanding shares	44
investment trust	35
retail sales	33
profit forecast	31
ordinary shares	31
net loss	30
third party	30
wholly-owned subsidiary	29
outstanding stock	29
average rate	27
net cash	27
annual meeting	26
net proceeds	26
public company	25
natural gas	25
gross product	24
gross domestic product	24
tax rate	24
cash flow	22
operating income	21
private placement	20
purchase price	19
proxy statement	18

Table 4.3: Financial lexicon hits

4.2 Precision

We ranked the extractions according to their confidence scores and examined the top 150 extractions and manually labeled them. The ratio of the correct extractions in both the top 50, *Precision@50* and the top 100 *Precision@100*, is 78% and the precision@150 is 79.33%. We examine the effect of different stages of the pipeline on the precision in section 5. The pipeline in [3], on the other hand, has a much lower precision (34%) for their *strict* supervised economic event and attribute extraction, i.e. where the financial value and event date have to match exactly, at 20% recall and 35% F1 score.

4.3 Knowledge Graph Statistics

We build a knowledge graph out of the extracted n-ary tuples. Each n-ary relation is decomposed into $\binom{n}{2}$ binary relations identified by *predicate.sense-id* (ex. *announce.01-0.8926653297234465*), where *id* is a random floating point number in the range [0.0, 1.0] generated for the n-ary relation. Figure 4.24 shows the relation **acquire.01**(*agent*: Merck, *thing acquired*: Cubist Pharmaceuticals Inc., *AM-TMP*: 04/04/2015) decomposed into 3 relations **acquired_by_0**: (*thing acquired*: Cubist Pharmaceuticals Inc., *agent*: Merck), **acquired_in_0**: (*thing acquired*: Cubist Pharmaceuticals Inc., *AM-TMP*: 04/04/2015), **acquired_in_0**:(*agent*: Merck, *AM-TMP*: 04/04/2015) where the suffix *0* is the relation *id*. Attaching an identifier ensures lossless decomposition of the relation and helps us identify its different arguments. Table 4.4 shows the knowledge graph statistics. The graph has over 31,000 weakly connected components (WCC), i.e subgraphs where each pair of nodes is connected by some path ignoring edge directions, and the diameter, i.e the longest distance between a pair of vertices, of the largest WCC is 19.

Number of nodes	248,923
Number of edges	380,079
Average degree	1.5269
Number of Weakly Connected Components	31,144
Diameter of the largest weakly connected component	19

Table 4.4: Knowledge graph statistics

4.4 Predicate and Patterns Distributions

In this section, we look at the distributions of the financial predicates, semantic frames and patterns in the knowledge graph and examine some extractions in more details.

Figure 4.1 shows the distribution of the top 20 predicates in SRL extracted facts while Figure 4.2 shows the distribution of the semantic frames over *all* extractions. Table 4.5 shows the different semantic frames and the predicates that evoke each frame.

The predicate *announce.01* which belongs to the semantic frame *Statement* makes up 17% of the total SRL extracted facts as shown in Figure 4.1. The semantic frame *Cause_change_of_position_on_a_scale* which includes the predicates such as 'increase.01','rise.01', 'fall.01' and 'decline.01' dominates with over 28% of the SRL extracted facts. Figures 4.5 to 4.22 show sample facts from the different semantic frames. The second example in Figure 4.6 demonstrates the effectiveness of the pipeline in extracting predicate-argument structures from sentences with long range dependencies (***Orb Energy*** *, an Indian solar company backed by U.S. venture capital fund Acumen Fund Inc ,* ***secured $ 10 million*** *in OPIC financing last year* ***for commercial rooftop projects.***). It also demonstrates an instance where the overly specific argument *commercial rooftop projects* is minimized by dropping the token *commercial* as none of its sub sequences is found in the financial lexicon. Figure 4.5 shows the pipeline successfully identifying and classifying the roles of different arguments of the predicate *increase.01*, i.e, '*causer of increase*', '*thing increasing*', '*start point*', '*end point*', '*amount increased by*','*AM-TMP*' in two different sentences. Such granularity is useful in answering complex queries such as one that seeks a contiguous sequence of stock prices that are all increasing (or decreasing). Figure 4.8, 4.20, 4.17 the correct classification of uncommon predicate senses in difficult contexts, *settle.02*, *cut.02* and *lose.02*. The correct classification of the predicate sense is the first step in assigning the correct semantic roles to its arguments.

On the other hand, the example in Figure 4.6 highlights a limitation of the coordinating conjunction processing. In this example, *hybrid sorghum and sunflower seed business* is a *combinatory* coordinating conjunction that should not be resolved. The coordinating conjunction processing could be extended with keywords indicating *combinatory* coordinating conjunction to prevent processing such conjunctions.

Figure 4.3 shows the patterns distribution. The most common pattern extracted is the *Demonym-ORG compound relational noun* followed by *Demonym-PER compound relational noun* and then *ORG-PER compound relational noun*. Table 4.6 shows the most common pattern extracted facts. President(Donald Trump, Unit-

edStates), which is an instance of the pattern *Demonym-PER compound relational noun*, was extracted 1026 times and is the most pattern extracted fact. All three patterns account for more than 90% of the pattern extracted facts. Figure 4.4 shows the top 20 relational nouns extracted via patterns. Over 40 presidents and over 25 CEOs were extracted. More of these pattern extracted facts are reported in Table 4.7.

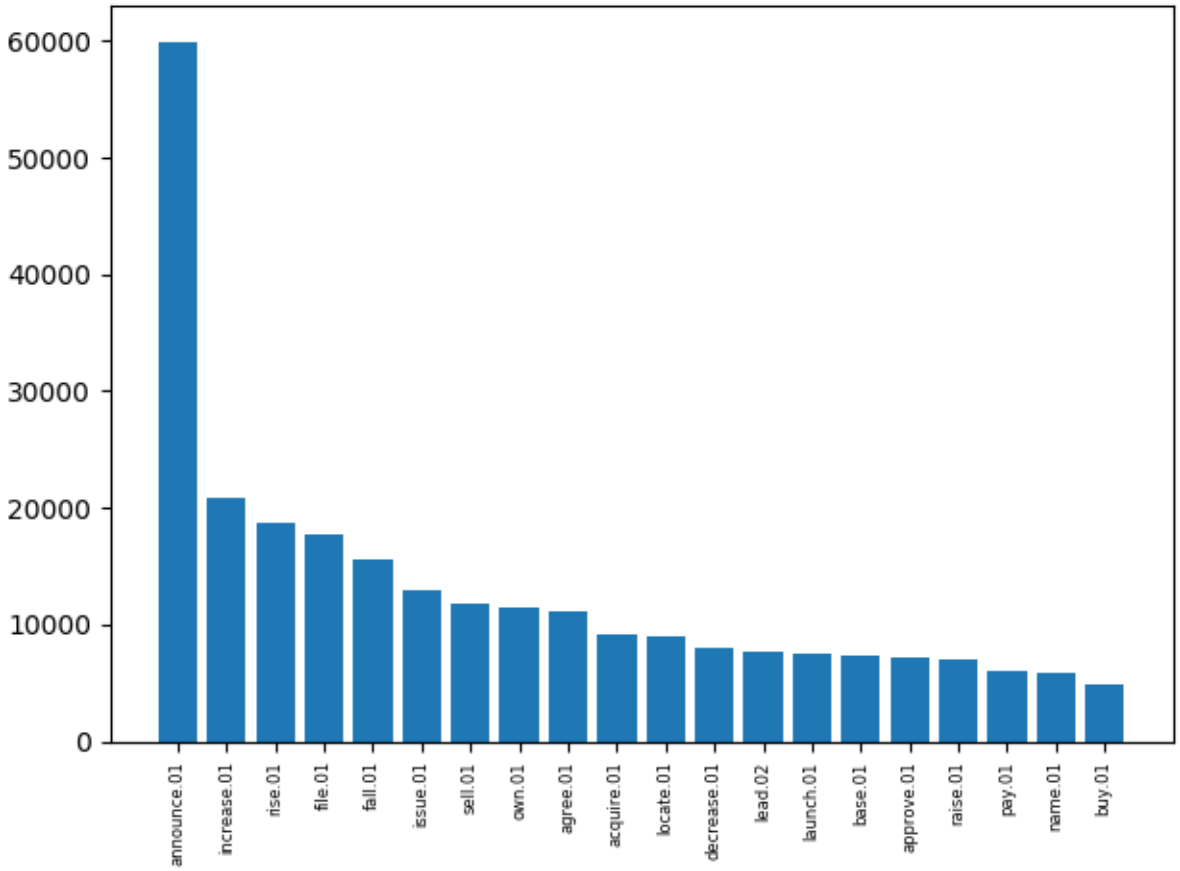


Figure 4.1: Predicate distribution

Frames	Predicates
Getting	['acquire.01', 'secure.01']
Commerce_buy	['buy.01', 'purchase.01']

Make_agreement_on_action	['agree.01', 'settle.02']
Statement	['announce.01', 'proclaim.01', 'declare.01']
Appointing	['appoint.01', 'name.01', 'name.03', 'designate.01', 'accredit.01']
Deny_or_grant_permission	['approve.01', 'authorize.01', 'sanction.01', 'sanction.02', 'grant.01']
Commerce_sell	['auction.01', 'sell.01', 'retail.01']
Change_position_on_a_scale	['balloon.02','climb.02', 'decline.01', 'decrease.01', 'drop.01', 'fall.01', 'grow.01', 'gain.01', 'increase.01', 'jump.01', 'rise.01', 'rocket.01', 'sink.01', 'skyrocket.01', 'reach.01', 'plummet.01', 'soar.01', 'surge.01', 'tumble.01', 'plunge.01', 'hit.01', 'rally.02', 'rebound.01', 'retreat.01', 'surpass.01', 'slip.01', 'slump.01']
expansion	['shrink.01', 'expand.01', 'inflate.01']
Cause_change_of_position_on_a_scale	['cut.02', 'slash.02', 'reduce.01', 'raise.01', 'lift.01']
Leadership	['direct.01', 'lead.02']
Process_end	['end.02', 'close.02'*]
Intentionally_create	['found.01']
Funding	['finance.01', 'fund.01', 'invest.01']
Activity_stop	['discontinue.01', 'terminate.01', 'shut.01']
Supply	['issue.01']

Activity_start	['launch.01']
Earnings_and_losses	['lose.02', 'earn.01']
Possession	['own.01']
Commerce_pay	['pay.01']
Using_resource	['spend.01']
Cause_change_of_strength	['strengthen.01', 'weaken.01']
economic_value	['worth.01', 'appreciate.01']
Legal_action	['sue.01']
Exporting	['export.01']
Importing	['import.01']
Locating	['base.01'*, 'locate.01']
Expensiveness	['cost.01']
Choosing	['elect.01']
Submitting_documents	['file.01']

Table 4.5: Frames and predicates

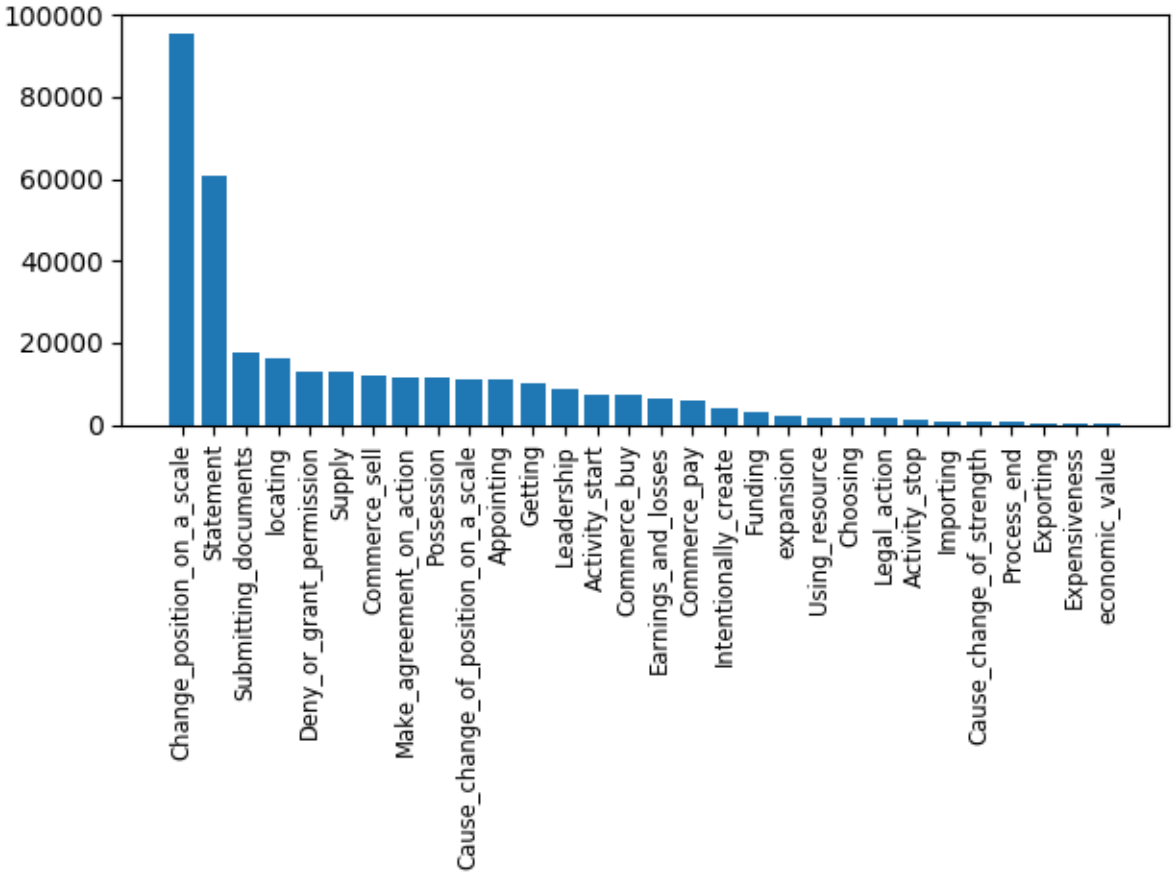


Figure 4.2: Frames distribution

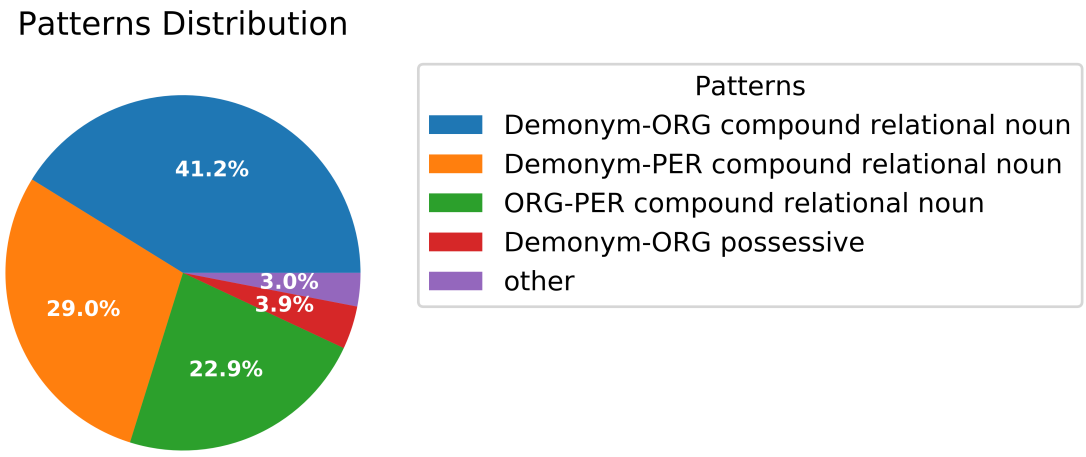


Figure 4.3: Pattern distribution

Fact	**Count**
President(Donald Trump, United States)	1026
President(Emmanuel Macron, France)	92
Secretary(Steven Mnuchin, Treasury)	81
Chancellor(Angela Merkel, Germany)	71
President(Xi Jinping, China)	63
Secretary(Steven Mnuchin, U.S. Treasury)	60
President(Vladimir Putin, Russia)	58
Secretary(Wilbur Ross, Commerce)	40
Prime Minister(Shinzo Abe, Japan)	38
Prime Minister(Theresa May, United Kingdom)	34

Table 4.6: The top ten most common pattern extracted facts

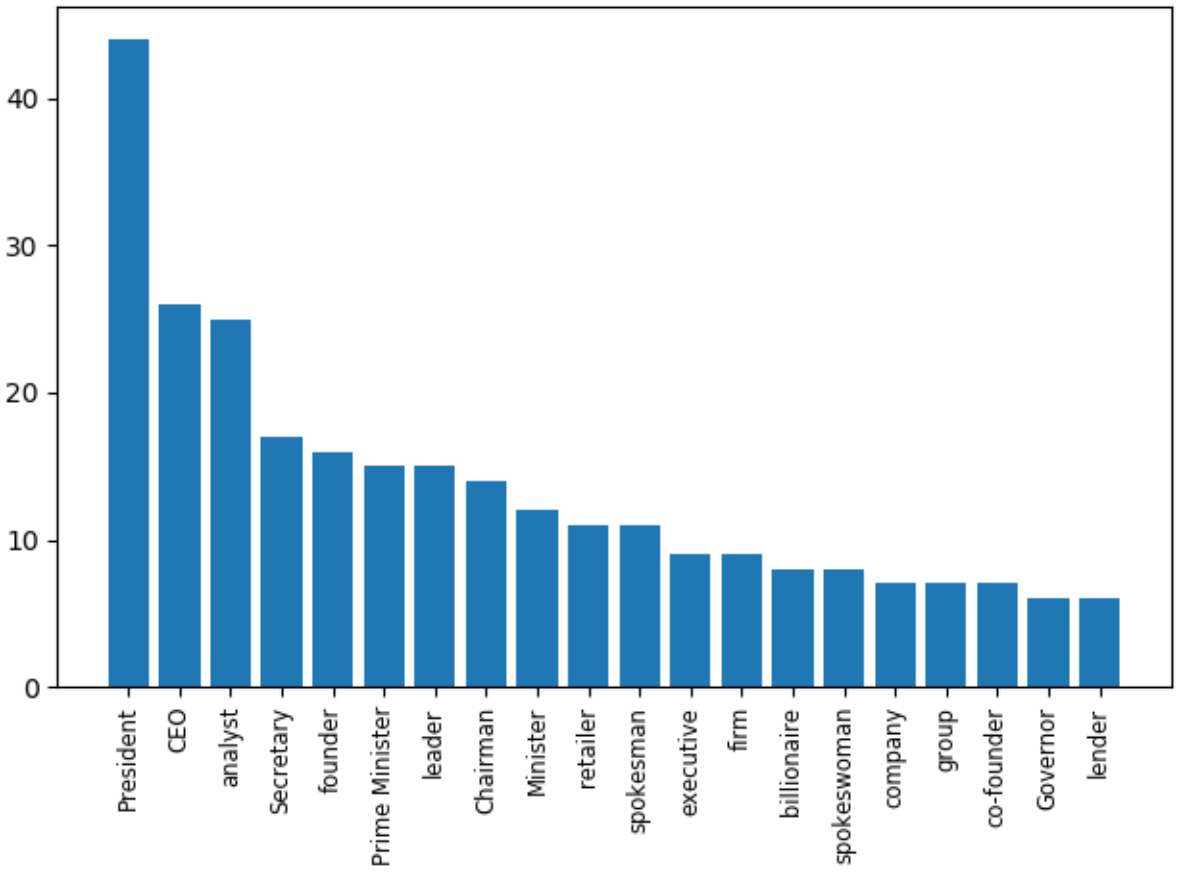


Figure 4.4: Top 20 relational nouns

- **Sentence**: Parke Bancorp's net loans increased to $ 1.01 billion at December 31 , 2017 , from $ 852.0 million at December 31 , 2016 , an increase of $ 159.8 million or 18.8 % .
- **Fact**: **increase.01**('*thing increasing*': "Parke Bancorp 's loans", '*start point*': '$ 852.0 million', '*end point*': '$ 1.01 billion', 'AM-TMP': '12/31/2017')
- **Sentence**: In addition , Comcast announced that Comcast increased Comcast dividend by 21 % to $ 0.76 per share on an annualized basis for 2018 .
- **Fact**: **increase.01**('*causer of increase*': 'Comcast', '*thing increasing*': 'Comcast dividend', '*amount increased by, EXT or MNR*': '21 %', '*end point*': '$ 0.76 per share')

Figure 4.5: Sample facts from the frame *Change_position_on_scale*

- **Sentence**: S&W Seed Company acquired S&W Seed Company hybrid sorghum and sunflower seed business from SV Genetics , a private Australian company specializing in the breeding and licensing of proprietary hybrid sorghum and sunflower seed germplasm , in May 2016.
- **Fact**: **acquire.01**('*agent, entity acquiring something*': 'S&W Seed Company', '*thing acquired*': 'S&W Seed Company hybrid sorghum business', '*seller*': 'SV Genetics', 'AM-TMP': '05/29/2016')
- **Sentence**: Orb Energy , an Indian solar company backed by U.S. venture capital fund Acumen Fund Inc , secured $ 10 million in OPIC financing last year for commercial rooftop projects.
- **Fact**: **secure.01**('*agent, entity acquiring something*': 'Orb Energy', '*thing acquired*': '$ 10 million', '*source, gained from*': 'rooftop projects', 'AM-TMP': '02/14/2017')

Figure 4.6: Sample facts from the semantic frame *Getting*

- **Sentence**: JAB bought Panera Bread Company in July 2017 through an investor group for $ 7.2 billion , expanding Panera Bread Company coffee and breakfast empire through the biggest-ever U.S. restaurant deal .
- **Fact**: **buy.01**:'*buyer*': 'JAB', '*thing bought*': 'Panera Bread Company', '*price paid*': '$ 7.2 billion', 'AM-TMP': '07/30/2017'

Figure 4.7: Sample facts from the semantic frame *Commerce_buy*

- **Sentence**: Hyundai settled a trade secrets lawsuit against Uber this year for $ 245 million in Uber equity , but not before a trade secrets lawsuit against Uber distracted Hyundai leaders and placed a black mark on Hyundai autonomous program .
- **Fact**: **settle.02**('*entity making resolution*': 'Hyundai', '*thing being resolved*': 'trade secrets lawsuit against Uber', '*value*': '$ 245 million', 'AM-TMP': '05/31/2018')

Figure 4.8: Sample facts from the frame *Make_agreement_on_action*

- **Sentence**: In 2006 , Steve Miley was appointed to the Senior Executive Service at NASA Headquarters in Washington , where he supervised the agency 's key technical capability portfolios as director of the Strategic Capabilities Assets Division .
- **Fact**: **appoint.01**('*appointee, employee*': 'Steve Miley', '*job, position*': 'Senior Executive Service', 'AM-TMP': '0 5 / 2 3 / 2 0 0 6', 'AM-LOC': 'NASA Headquarters in Washington')

Figure 4.9: Sample facts from the semantic frame *Appointing*

- **Sentence**: Chongqing Oil and Gas Exchange sold 28 million cubic meters of pipeline gas in debut trading .
- **Fact**: **sell.01**('*Seller*': 'Chongqing Oil', '*Thing Sold*': '28 million meters of pipeline gas')
- **Sentence**: Ford also auctioned off the 2019 Ford Mustang Bullitt with VIN 001 for $ 300,000 .
- **Fact**: **auction.01**('*seller*': 'Ford', '*sold*': '2019 Ford Mustang Bullitt', '*Price Paid*': '$ 300,000')

Figure 4.10: Sample facts from the semantic frame *Commerce_sell*

- **Sentence**: Ghana 's central bank cut Ghana 's central bank benchmark interest rate by 100 basis points to 17 percent on Monday , saying Ghana 's central bank was on track to meet Ghana 's central bank medium-term inflation target as the economy stabilised .
- **Fact**: **cut.02**('*cutter*': "Ghana 's central bank", '*thing reduced*': "Ghana 's benchmark interest rate", '*amount reduced by*': '100 basis points', '*end point*': '17 percent', 'AM-TMP': '0 5 / 1 4 / 2 0 1 8')

Figure 4.11: Sample facts from the frame *Cause_change_position_on_scale*

- **Sentence**: Emerging Market Media was founded December 2015 by Dawn Kissi , an award-winning international journalist , who began her career in the Network newsroom of ABC News .
- **Fact**: **found.01**('*agent, setter*': 'Dawn Kissi', '*thing set*': 'Emerging Market Media', 'AM-TMP': '1 2 / 0 2 / 2 0 1 5')

Figure 4.12: Sample facts from the semantic frame *Intentially_create*

- **Sentence**: Hudson financed the Kaufhof acquisition by using a joint venture that acquired Kaufhof 's real estate and became Hudson 's Bay landlord .
- **Fact**:**finance.01**('*financ(i)er*': 'Hudson', '*thing financed*': 'Kaufhof acquisition', '*money*': 'using joint venture')

Figure 4.13: Sample facts from the semantic frame *Funding*

- **Sentence**: Acorda Therapeutics also recently discontinued the development of Acorda Therapeutics Parkinson 's drug after reports of death .
- **Fact**: **discontinue.01**('*entity ending something, agent*': 'Acorda Therapeutics', '*thing discontinued*': "development of Acorda Therapeutics Parkinson 's drug")

Figure 4.14: Sample facts from the semantic frame *Process_end*

- **Sentence**: During the quarter ended March 31 , 2018 , A&S issued $ 1.25 billion of unsecured debt at a cost of funds of 6.875 % .
- **Fact**: **issue.01**('*issuer*': 'A&S', '*thing issued*': '$ 1.25 billion of unsecured debt', '*attribute, issued as or at*': ' at a cost of funds of 6.875 %')

Figure 4.15: Sample facts from the semantic frame *Supply*

- **Sentence**: In April 2018 , Inovalon launched two new cloud-based offerings on the Inovalon ONE Platform.
- **Fact**: **launch.01**('*starter, agent*': 'Inovalon', '*thing being launched*': 'two new cloud-based offerings', 'AM-TMP': '0 4 / 0 9 / 2 0 1 8')

Figure 4.16: Sample facts from the semantic frame *Activity_start*

- **Sentence**: Emaar , which has lost 10 percent so far this year because of a weak local real estate market , has been a big drag on Qatar 's benchmark index
- **Fact**: **lose.02**('*entity losing something*': ' Emaar', '*thing lost*': ' 10 percent', 'AM-TMP': '04/16/2018')

Figure 4.17: Sample facts from the semantic frame *Earnings_and_losses*

- **Sentence**: GE spent almost $ 80 billion buying back shares at prices over $ 30 .
- **Fact**: **spend.01**('*spender, buyer*': 'GE', '*thing bought, commodity*': 'buying back shares at prices over $ 30', '*price paid, money*': 'almost $ 80 billion')

Figure 4.18: Sample facts from the semantic frame *Using_resource*

- **Sentence**: Walmart paid $ 3.3 billion for internet retailer Jet.com and Walmart innovative pricing software in the August of 2016 here .
- **Fact**: **pay.01**('*payer or buyer*': 'Walmart', '*money or attention*': '$ 3.3 billion', '*commodity, paid for what*': 'internet retailer Jet.com', 'AM-TMP': '08/10/2016')

Figure 4.19: Sample facts from the semantic frame *Commerce_pay*

- **Sentence**: Mexico 's peso and Russia 's rouble strengthened 0.5 percent .
- **Fact**: **strengthen.01**('*causal agent*': "Mexico 's peso", '*thing strengthening*': '0.5 percent')

Figure 4.20: Sample facts from the frame *Cause_change_of_strength*

- **Sentence**: HRG Group appreciated by 9 % in 2017 to $ 16.95 per share at year-end .
- **Fact**: **appreciate.01**('*thing increasing*': 'HRG Group', '*EXT*': '9 %', 'AM-TMP': '0 2 / 2 3 / 2 0 1 7')

Figure 4.21: Sample facts from the semantic frame *economic_value*

- **Sentence**: Corephotonics sued Apple in November for violating four dual-camera patents Corephotonics holds in Corephotonics iPhone7 Plus , the first Apple handset to ship with a dual-lens camera .
- **Fact**: **sue.01**('*litigant*': 'Corephotonics', '*defender*': 'Apple', '*crime*': 'violating four dual-camera patents', 'AM-TMP': 'in November')

Figure 4.22: Sample facts from the semantic frame *Legal_action*

Relation	First argument	Second argument
PresidentOfCountry	Donald Trump	United States
PresidentOfCountry	Emmanuel Macron	France
PresidentOfOrg	Mario Draghi	European Central Bank
PresidentOfCountry	Vladimir Putin	Russia
PresidentOfCountry	Tayyip Erdogan	Turkey
PresidentOfOrg	Donald Tusk	European Council
PresidentOfCountry	Abdel Fattah al-Sisi	Egypt
PresidentOfCountry	Bashar al-Assad	Syria
PresidentOfCountry	Raul Castro	Cuba
PresidentOfCountry	Barack Obama	United States
PresidentOfCountry	Pedro Pablo Kuczynski	Peru
PresidentOfCountry	Jimmy Morales	Guatemala
PresidentOfCountry	Joko Widodo	Indonesia
PresidentOfOrg	Katherine Fedor	NBRC
PresidentOfOrg	Akio Toyoda	Toyota Motor

founder	Niki Lauda	Niki
founder	Ingvar Kamprad	IKEA
founder	Julian Assange	WikiLeaks
founder	Luciano Benetton	companyBenetton Group
founder	Jeremy Tooker	Barrel Coffee
founder	Jack Ma	Alibaba
founder	Gates	Microsoft
founder	Jamie Heywood	PatientsLikeMe
founder	Patrick Drahi	Altice
founder	Anya Fernald	California Belcampo
founder	Ray Dalio	Bridgewater Associates
founder	Jia Yueting	LeEco
founder	Emily McDowell	EMS
founder	Tim Draper	Draper Associates
founder	Jack Penrod	Nikki Beach Worldwide
founder	David T. Wilentz	Wilentz
BillionaireFromCountry	Herz	Germany
BillionaireFromCountry	Mohammed al-Fayed	Egypt
BillionaireFromCountry	Vincent Bollore	France
BillionaireFromCountry	Richard Branson	United Kingdom
BillionaireFromCountry	Mikhail Fridman	Russia
BillionaireFromCountry	Wang Jianlin 's	China
BillionaireFromCountry	Viktor Vekselberg	Russia
CEO	Ronny Shmoel	Circuit City
CEO	Johan Lundgren	easyJet
CEO	Ed Bastian	Delta
CEO	Jamie Dimon	J.P. Morgan Chase
CEO	Jon Brod	Confide
CEO	Jia Yueting	LeEco
CEO	Steve Jobs	Apple
CEO	Kenneth Chenault	American Express
CEO	Mallon	Ulster Bank
CEO	Bob Senior	Thibaut Inc. 's
CEO	Gregory Scott	New York & Company 's
CEO	Patrick Doyle	Domino 's Pizza
CEO	Steve Wynn	The Wall Street Journal
CEO	John Landgraf	FX Networks

CEO	Cliff Hudson	Sonic
CEO	James Connelly	Fetch
CEO	William Farrow	Urban Partnership Bank
CEO	Eric Schmidt	Google
CEO	John Phillips	PredictIt
CEO	Evan Spiegel	Snap
CEO	Bill McDermott	SAP SE
CEO	Jack Dorsey	Twitter
CEO	Mark Zuckerberg	Facebook
DrugmakerFromCountry	Sanofi	France
DrugmakerFromCountry	Merck	Germany
DrugmakerFromCountry	Dermapharm	Germany

Table 4.7: Sample relations extracted via patterns

4.5 Ablation Studies

We conduct ablation studies on a sample of 1000 articles to demonstrate the effect of each stage of the pipeline on the overall performance. Table 4.8 shows the ablation studies statistics. The first column reports the statistics of running all stages of the pipeline on the sample. The effects of turning off different stages of the pipeline are reported in the corresponding columns. We label the top 100 extractions to measure the precision.

Turning off the cleaning module results in having noisy text spans that make sentences much longer and end up being filtered in the sentence length filtering stage. This results in fewer overall extractions (48.5% drop). More importantly, the precision significantly drops in the top 50 and top 100 extractions by 12% and 4% respectively.

Turning off the co-reference stage yields more sentences. This results from having shorter sentences,i.e. where co-references are not resolved, that pass the sentence length filtering stage. Turning off this stage did not affect the number of the pattern extracted facts which is not surprising since these patterns rarely occur in contexts where resolution is needed. However, fewer facts were extracted via SRL and more appositions were extracted. A significant increase of 4% in precision can be seen at the top 50 extractions. This suggests we can improve the overall precision without sacrificing much of the *recall* (only 4.2% drop) of the system by turning off the co-reference stage. More tokens were dropped by the minimization module as a result of having more appositions.

Turning off the financial predicate filtering stage results in 18.2% increase in the total number of facts. However, that comes at a 68% loss in precision at the top 50 extraction. It's worth noting that we still have financial argument validation in place and hence, financially relevant facts should not suffer a drop in precision. This suggests that more non relevant facts have higher confidence scores. Since the fact scoring model was only trained on financially relevant facts, this could be the reason behind this performance.

Turning off the coordinating conjunctions stage results in a 2% increase in precision@50 at a 16.5% drop in the overall number of extractions. Turning off appositions extraction yields a 2% increase in precision@50 and 4% increase in precision@100 at a 53% drop in the overall number of extracted facts.

So far the average argument length has not been much affected except when turning off the appositions extraction which saw a 34.2% increase. This suggests that average argument length of SRL extracted facts is much higher. When turning off the minimization module, we can see a 13.1% increase in the average argument length. This indicates the significance of this module in minimizing overly specific arguments while preserving financially relevant parts.

There are a number of insights that this study has provided. The financial predicate filtering had shown to be the most important factor in precision. Furthermore, the cleaning stage presents a significant trade off in precision and recall (turning it off results in 12% drop in precision at a 48.5% increase in recall). Depending on the downstream applications, we may want to favor precision over recall or vice versa. This study highlights the stages that play a role in either of the metrics.

Metric	**None**	**Cleaning**	**Co-reference resolution**	**Financial Predicate Dictionary Filtering**	**coordinating conjunc-tions**	**Appositions**	**Minimization**
Number of sentences processed	13101	6521	13144	13101	13101	13101	13101
Number of sentences eliminated	12379	6158	12455	6578	12379	12651	12379
Percentage of sen-tences with extracted facts	5.51%	5.56%	5.24%	58.7%	5.51%	3.43%	5.51%
Number of Temporal arguments parsed to DATE	184	94	184	3384	181	184	184
Number of distinct pattern extracted facts	24	20	24	611	24	15	31

Number of SRL extracted facts	590	305	531	18839	493	585	590
Number of Appositions	665	333	670	5118	550	N/A	665
Average argument length	3.80	3.83	3.75	3.56	3.96	5.10	4.30
Number of Tokens dropped by minimiza-tion	1293	628	1397	24502	1215	711	N/A
P@50	82%	70%	86%	14%	84%	84%	80%
p@100	79%	75%	80%	32%	80%	83%	77%
Percentage of domain relevant facts	100%	100%	100%	14.1%	100%	100%	100%
Percentage of domain relevant facts in the top 100	100%	100%	100%	36%	100%	100%	100%

Table 4.8: Ablation studies

4.6 Knowledge Graph Querying and Subgraph Visualization

One of the advantages of knowledge graphs is that they can be conveniently stored and queried in a high level declarative language such as Datalog. We demonstrate this capability by showing 3 Datalog queries that we ran against the knowledge graph. These queries demonstrate the tradeoff between precision and recall. While the answers to those queries is correct, the recall is limited and in certain instances, particularly the first query, achieving high recall is just as important as precision.

- The trade between the United States and China: in this query, we are asking for the exports between the two countries.

  export(product):- **export**(”United States”, product, ” China”)

  export(product):- **export**(”China”, product, ”United States”)

  Figure 4.23 shows the resulting subgraph.

- Acquisitions by German drugmakers: In this query, we ask for the companies that were acquired by German drugmakers. This is a join query over the *Demonym-ORG compound relational noun* pattern and the *acquire.01* predicate argument structures.

  Acquisition(drugmaker, acquired_company) :- **acquire.01**(drugmaker, acquired_company), **pattern**(drugmaker, “is a drugmaker of/from”, “Germany”)

  Figure 4.24 shows the resulting subgraph.

- Companies suing each other on patent related grounds: This is a query over the *sue.01* predicate argument structures.

  sued(litigator, defender) :- **sue.01**(litigator, defender, “%patent%”)

  Figure 4.26 and 4.26 show the resulting subgraphs.

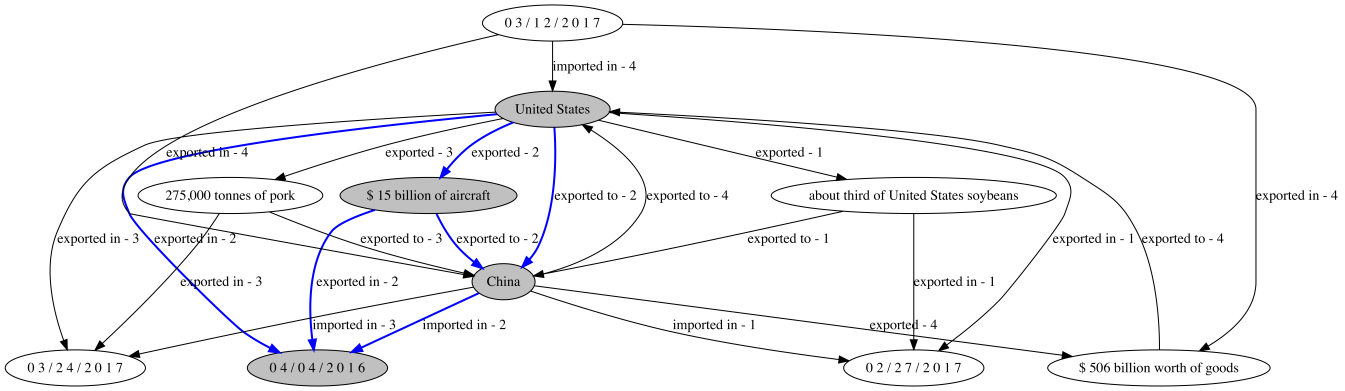


Figure 4.23: US-China trade. The relation highlighted shows that the *United States* exported *$15 billion dollar of aircraft* to *China* on *04/04/2016*. Section 4.4 discusses how we break down n-ary relations to create the knowledge graph.

→ fact based on what was reported in news

→ no verification / fact-checking taking

→ may be incomplete / incorrect

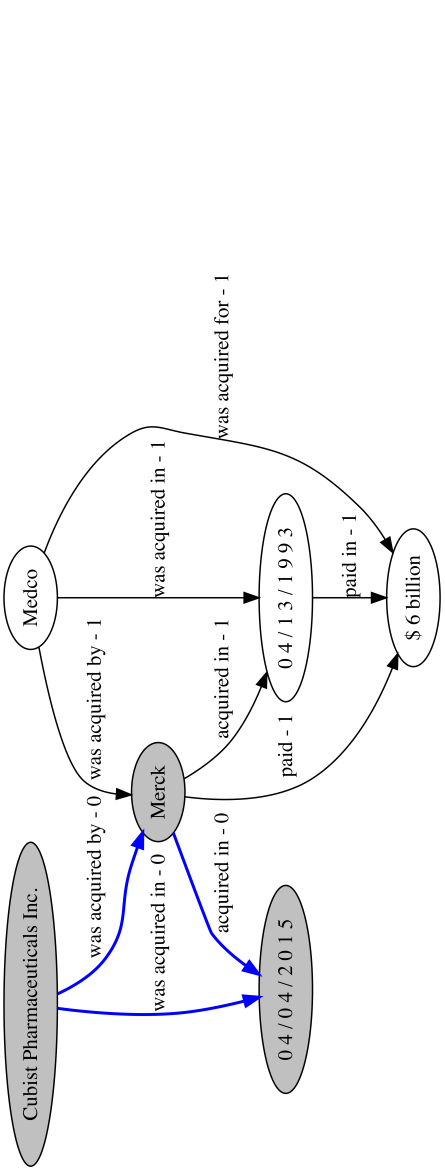


Figure 4.24: Subgraph representing acquisitions by German drugmakers. The relation highlighted shows that the *Merck* acquired *Cubist Pharmaceuticals Inc.* on *04/04/2015*.

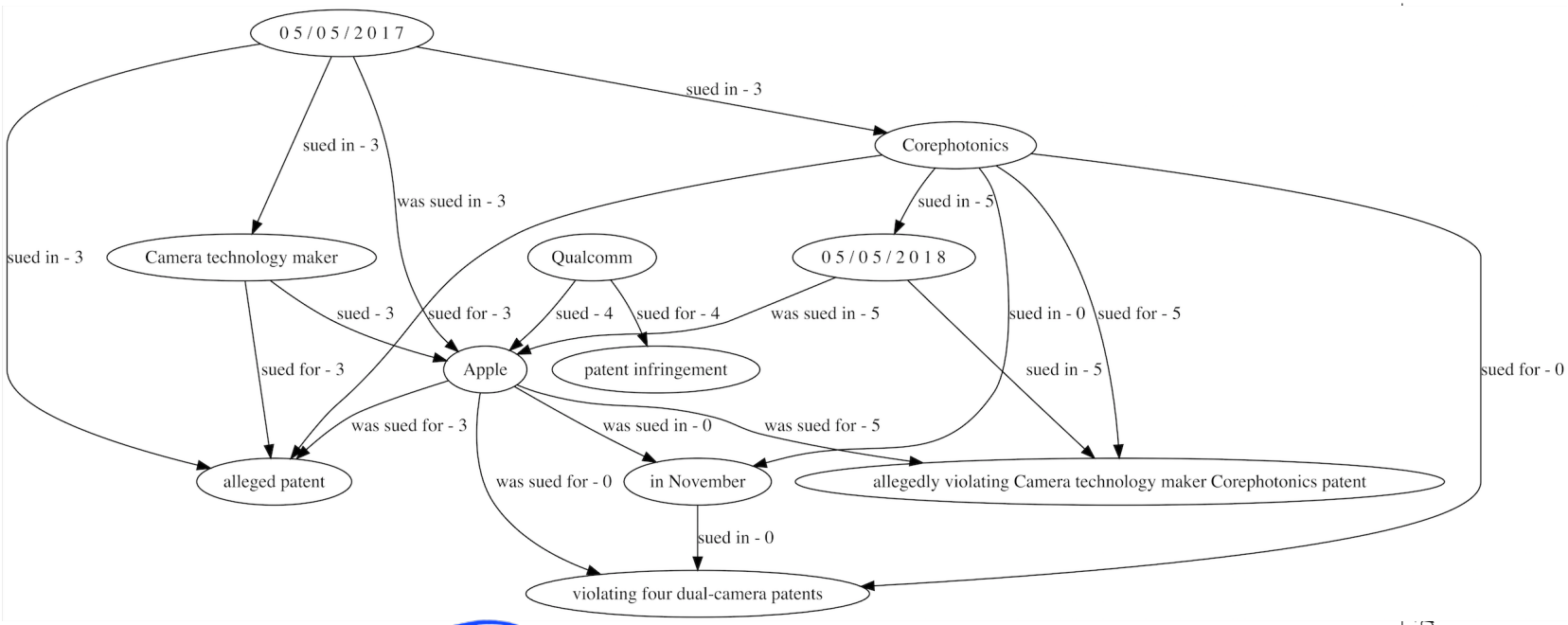


Figure 4.25: Companies suing on patent grounds.

→ visualization seems too complex !!!

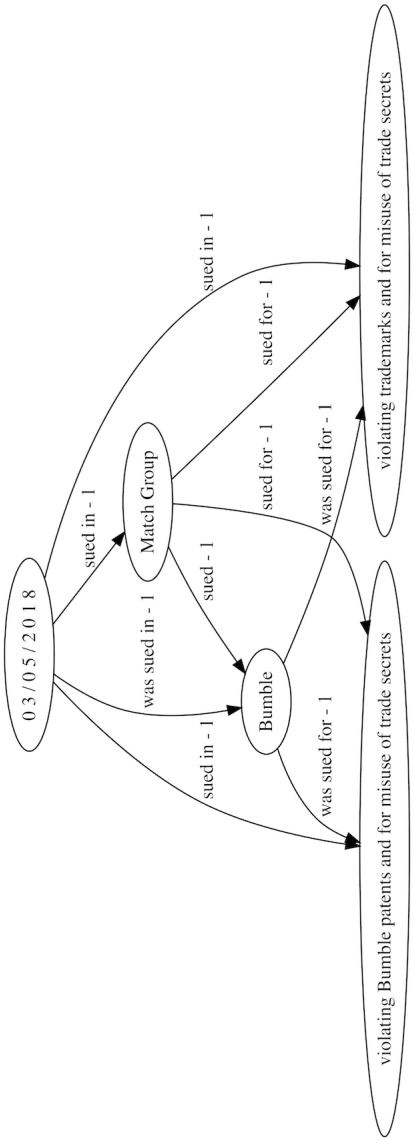


Figure 4.26: Companies suing on patent grounds.

Chapter 5

Discussion and Future Work

In section 4, we presented the results of running the pipeline over ∼300,000 financial news articles and extracted ∼342,000 facts from only 5.2% of the sentences at 78% precision@100. One way to extract more facts is by expanding the financial predicate dictionary. This, however, may come at a potential loss in precision as we highlighted in the ablation studies but it's worth noting that turning off the financial predicate filtering is not the same as expanding the dictionary. Semantic and structural constraints on the new predicates could be introduced to help improve the quality of the extractions. This, however, may not be a significant factor since we learnt that only 1.52% of the relations were excluded as a result of failing the predicate argument validation.

The ablation studies also helped uncover the trade off between precision and recall that different stages presented and suggested that the co-reference resolution system could be turned off to improve the precision without sacrificing a significant drop in recall. Indeed, we found many instances where the co-reference resolution makes seemingly unrelated associations. Figure 5.1 shows an excerpt from a news article that undergoes co-reference resolution. The system resolves *KKR portfolio company* in the sentence *WebMD was acquired by a KKR portfolio company for $ 2.8 billion in 2017* to the drugmaker *Merck* that is mentioned only once 2 sentences above. The system at the same time fails to make the association between *Wygod* and *his* in the sentence *Under his direction ....*

We believe, however, that the continued advances in neural co-reference resolution such as SpanBERT [14] will lead to better performing systems that will help lift the precision without sacrificing the recall at all. The studies highlighted the significance of the cleaning stage in the overall precision and the importance of resolving coordinating conjunctions and appositions in generating more facts without heavily impacting the precision. However, the limitations with the coordinating conjunctions processing described in sections 3.8 and 4.4 could be addressed in future work.

While the pattern extraction stage contributed to only 1.3% of the facts, they were high precision facts owing to typed patterns that are commonly found in the financial domain. We would like to explore mining additional patterns from the corpus using a seed set and generalizing over the extracted facts similar to the

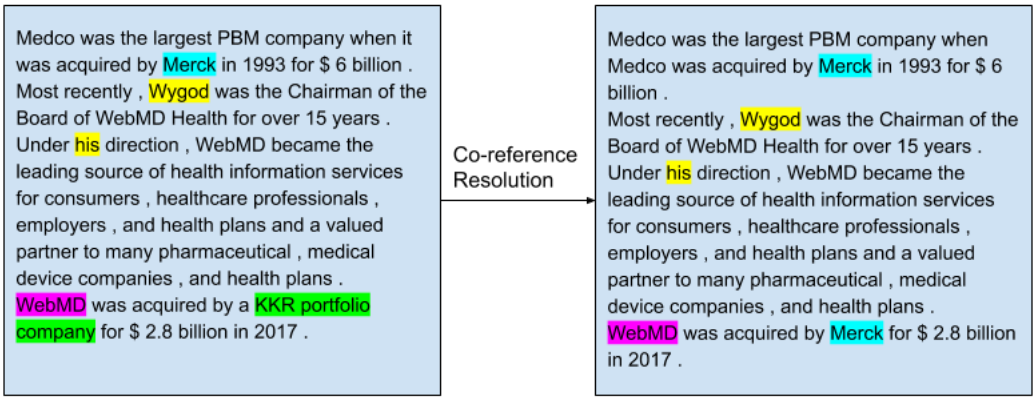


Figure 5.1: Co-reference resolution errors.

work in [32]. We also would like to examine ideas for scoring pattern related facts that combines our confidence in patterns reflected by the number of unique facts it extracted and confidence in facts reflected by the number of unique patterns that extracted it.

Although the temporal parser succeeds in parsing 78.5% of the temporal arguments. It has a limitation that is highlighted in Figure 4.6. In the first sentence, *May 2016* is resolved to *05/29/2016* relative to the publication date. Figure 4.25 shows another instance where the *this week* in the sentence *Camera technology maker Corephotonics this week sued Apple for allegedly violating Camera technology maker Corephotonics patent related to dual cameras in smartphones* is resolved to *05/05/2018*. Ideally such instances should be resolved to only month and year.

The ablation study has also shown the importance of the minimization stage in decreasing the average argument length. We would like to examine expand-

ing the minimization beyond the safe and dictionary minimization of adverbial patterns. Prepositional phrases, although good candidates for minimization, are equally challenging. One example is shown in Figure 4.9, the location argument *NASA Headquarters in Washington* could be minimized to just *NASA Headquarters* or broken down into 2 arguments *NASA Headquarters* and *in Washington*. The argument *trade secrets lawsuit against Uber* in Figure 4.8 could similarly be broken down. We would like to explore ways to minimize such instances without trading the precision. Figure 4.14 shows an instance where the *agent* is co-referenced in the *patient*. This makes the argument unnecessarily overly specific since it's implied that *Parkinson 's drug* is developed by *Acorda Therapeutics*. Such instances could be made less specific.

Along these lines, it is common to see arguments such as *more than 3%*, *as much as 3 percent*, *almost 3%* when reporting stocks rising and falling. We would like to *canonicalize* these arguments in to $< 3\%$ or bin them into numeric ranges. We believe this would maximize the utility of these arguments in downstream applications.

Many of the facts do not have temporal arguments. Figure 4.20 shows an instance where this information is critical. Currencies strengthen and weaken constantly and it's important to provide the time frame. One idea is to always attach the publication date to such instances or only include instances where this information is available.

While the pipeline is targeted towards the financial domain, it can be transferred to a different domain with the following adjustments:

- A domain targeted dictionary for filtering candidate predicate argument structures.
- A domain targeted lexicon for dictionary minimization of overly specific arguments.
- Supervised models for cleaning and fact scoring trained on target domain datasets

We would like to examine the *transferability* of the pipeline to other domains in future work.

We would also like to explore methods to build domain targeted dictionary using corpus level statistics and/or learn models that incorporate consistency constraints (e.g., domain, equality, mutual exclusion, type signature) and automatically identify fact relevance using triple level statistical relevance features. The Probabilistic Similarity Logic (PSL) [27], [5] modeling framework could ideally be used in this scenario. We would also like to address the case when a financial event is

reported across multiple articles with partial argument identification. One way to address this is to incorporate consistency constraints that help in unifying partial facts reporting the same event into full n-ary facts.

In section 4.4, we described how we decompose n-ary relations into $\binom{n}{2}$ binary relations while preserving information. Another way is RDF reification [10] which introduces an identifier to the primary fact and adds subsequent triples for subject, predicate, object and other properties. While RDF reification was widely adopted in may KB like YAGO, it suffers issues with consistency and adds overhead to query processing. Other approaches like subproperties [23], and Neo-davidsonian representations [15] have been proposed to address the overhead of RDF reification. We would like to explore the best approach for representing n-ary relations.

Chapter 6

Conclusions

In this work, we develop a high precision pipeline for knowledge extraction from financial news corpus. The pipeline produces over 340,000 verb and noun mediate relational tuples at 78% precision at the top 100 extractions by employing semantic role labeling and pattern based information extraction. We build a financial predicate dictionary that places structural and semantic constraints predicate argument structures helping produce high quality domain targeted extractions. To maximize the utility of the extractions in downstream applications, we minimize arguments that are considered overly-specific by processing coordinating conjunctions, appositions and employing financial lexicon to minimize adverbial nouns. We evaluate the pipeline and the resulting knowledge graph on a number of metrics and conduct ablation studies to examine the effect of the different modules of the pipeline on these metrics. These studies offered a number of insights and demonstrated the importance of both the financial predicate dictionary filtering and the noisy text cleaning stages in the overall precision of the pipeline. We demonstrated the querying capabilities of the resulting knowledge graph via Datalog queries and visualized the resulting subgraphs. In future work, we would like to explore the transferability of the pipeline to different domains, canonicalizing numeric arguments and mining domain targeted dictionary from the corpus for filtering noisy extractions.

Bibliography

[1] Collin F Baker, Charles J Fillmore, and John B Lowe. The berkeley framenet project. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*, pages 86–90. Association for Computational Linguistics, 1998.

[2] Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open information extraction from the web. In *IJCAI*, pages 2670–2676, 2007.

[3] Jan R Benetka, Krisztian Balog, and Kjetil Norvag. Towards building a knowledge base of monetary transactions from a news collection. In *2017 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, pages 1–10. IEEE, 2017.

[4] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250, 2008.

[5] M. Broecheler, L. Mihalkova, , and L Getoor. Probabilistic similarity logic. In *In Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*, pages 73—-82. AUAI Press, 2010.

[6] Kevin Clark and Christopher D Manning. Deep reinforcement learning for mention-ranking coreference models. *arXiv preprint arXiv:1609.08667*, 2016.

[7] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–610, 2014.

[8] Kiril Gashteovski, Rainer Gemulla, and Luciano del Corro. MinIE: Minimizing facts in open information extraction. In *Proceedings of the 2017*

Conference on Empirical Methods in Natural Language Processing, pages 2630–2640, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[9] José Gutiérrez-Cuellar and José Manuél Gómez-Pérez. Havas 18 labs: A knowledge graph for innovation in the media industry. In *International Semantic Web Conference (Industry Track)*, volume 1383, 2014.

[10] Patrick J Hayes and Peter F Patel-Schneider. Rdf 1.1 semantics. *W3C recommendation*, 25:7–13, 2014.

[11] Luheng He, Kenton Lee, Omer Levy, and Luke Zettlemoyer. Jointly predicting predicates and arguments in neural semantic role labeling. *arXiv preprint arXiv:1805.04787*, 2018.

[12] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. 2017.

[13] Richard Johansson and Pierre Nugues. Dependency-based semantic role labeling of propbank. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 69–78, 2008.

[14] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020.

[15] Daniel Jurafsky and James H Martin. Speech and language processing.

[16] Paul Kingsbury and Martha Palmer. From treebank to propbank. In *LREC*, pages 1989–1993. Citeseer, 2002.

[17] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[18] Jens Lehmann, Robert Isele, Max Jakob, Anja Jentzsch, Dimitris Kontokostas, Pablo N Mendes, Sebastian Hellmann, Mohamed Morsey, Patrick Van Kleef, Sören Auer, et al. Dbpedia–a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, 6(2):167–195, 2015.

[19] Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60, 2014.

[20] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.

[21] Bhavana Dalvi Mishra, Niket Tandon, and Peter Clark. Domain-targeted, high precision knowledge extraction. *Transactions of the Association for Computational Linguistics*, 5:233–246, 2017.

[22] Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bo Yang, Justin Betteridge, Andrew Carlson, B Dalvi, Matt Gardner, Bryan Kisiel, et al. Never-ending learning. *Communications of the ACM*, 61(5):103–115, 2018.

[23] Vinh Nguyen, Olivier Bodenreider, and Amit Sheth. Don't like rdf reification?: making statements about statements using singleton property. In *Proceedings of the 23rd international conference on World wide web*, pages 759–770. ACM, 2014.

[24] Naoaki Okazaki. Crfsuite: a fast implementation of conditional random fields (crfs), 2007.

[25] Harinder Pal et al. Demonyms and compound relational nouns in nominal open ie. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, pages 35–39, 2016.

[26] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proc. of NAACL*, 2018.

[27] Jay Pujara, Hui Miao, Lise Getoor, and William Cohen. Knowledge graph identification. In *Proceedings of the 12th International Semantic Web Conference - Part I*, ISWC '13, pages 542–557, New York, NY, USA, 2013. Springer-Verlag New York, Inc.

[28] Amit Singhal. Introducing the knowledge graph: things, not strings. *Official google blog*, 5, 2012.

[29] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706, 2007.

[30] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014.

[31] Xuan Wang, Yu Zhang, Qi Li, Yinyin Chen, and Jiawei Han. Open information extraction with meta-pattern discovery in biomedical literature. In *Proceedings of the 2018 ACM International Conference on Bioinformatics, Computational Biology, and Health Informatics*, pages 291–300, 2018.

[32] Mohamed Yahya, Steven Whang, Rahul Gupta, and Alon Halevy. Renoun: Fact extraction for nominal attributes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 325–335, 2014.