A High Precision Pipeline for Financial Knowledge Graph Construction

Sarah Elhammadi, Laks V.S. Lakshmanan, Raymond Ng, Michael Simpson
University of British Columbia, Vancouver, Canada
shammadi,laks,rng,mesimp@cs.ubc.ca

Baoxing Huai, Zhefeng Wang, Lanjun Wang*
Huawei Technologies Co. Ltd., Hangzhou, China
huaibaoxing,wangzhefeng,lanjun.wang@huawei.com

Abstract

Motivated by applications such as question answering, fact checking, and data integration, there is significant interest in constructing knowledge graphs by extracting information from unstructured information sources, particularly text documents. Knowledge graphs have emerged as a standard for structured knowledge representation, whereby entities and their inter-relations are represented and conveniently stored as (*subject*, *predicate*, *object*) triples in a graph that can be used to power various downstream applications. The proliferation of financial news sources reporting on companies, markets, currencies, and stocks presents an opportunity for extracting valuable knowledge about this crucial domain. In this paper, we focus on constructing a knowledge graph automatically by information extraction from a large corpus of financial news articles. For that purpose, 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. The extracted triples are stored in a knowledge graph making them readily available for use in downstream applications.

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)* (SPO) 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. Driven by applications such as question answering, fact checking, information search, data integration and recommender systems, there is tremendous interest in extracting high quality knowledge graphs by tapping various data sources (Ji et al., 2020; Noy et al., 2019; Dong et al., 2014; Shortliffe, 2012; Lehmann et al., 2015).

Over the years, a number of *cross domain* KGs have been created including DBpedia (Lehmann et al., 2015), YAGO (Suchanek et al., 2007), Freebase (Bollacker et al., 2008) and NELL (Mitchell et al., 2018), covering millions of real world entities and thousands of relations, across different domains. Recently, there has been a growing interest in generating *domain targeted* structured representations of financial and business entities and how they are related to each other. Crunchbase [1] curated a knowledge base (KB) through partnerships with companies and data experts covering 100,000+ business entities including companies and investors, but covering only a few types of business transactions (i.e., relations) such as acquisitions and funding rounds. The work by (Benetka et al., 2017) attempted to address this limitation by developing a pipeline to populate a KB semi-automatically with quintuples of the form *(subject, predicate, object, monetary value, date)* extracted from a news corpus. However, this pipeline

*Corresponding author.

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

Proceedings of the 28th International Conference on Computational Linguistics, pages 967–977
Barcelona, Spain (Online), December 8-13, 2020

only extracted 496 quintuples covering 316 economic events that fall into one of two categories: events that increment the value of agent's resources (e.g., acquire, collect) or decrement it (e.g., pay, sell).

Our goal is to automatically extract high precision structured representations from thousands of financial news articles covering a broader range of financial entities such as markets, stocks, persons (e.g., CEOs, presidents, etc), currencies, and governments. Further, storing them in a KG can facilitate answering interesting and complex queries such as (1) company acquisitions by German drugmakers, (2) US-China trade in terms of exports, (3) companies suing each other on patent grounds, etc.

Knowledge extraction from unstructured sources is one of the major challenges facing industry scale KGs (Noy et al., 2019). Traditional approaches to KG construction from text rely on a pre-specified ontology of relations and large amounts of human annotated training data to learn extraction models for each relation. This limits their scalability and applicability to new relation types. Open Information Extraction (OpenIE) (Banko et al., 2007) aims to overcome these limitations by extracting *all* semantic relational tuples in raw surface form with little or no human supervision. Closely related to OpenIE is Semantic Role Labeling (SRL) which aims at detecting argument structures associated with verb predicates, as well as labeling their semantic roles, thus overcoming situations where the verb tense and conjugation change the role of the argument in the sentence (i.e., whether the argument is an *agent* which carries out the predicate action or a *theme* which receives the predicate action). Semantic roles make it possible to impose *structural* and *semantic constraints* on entity types to ensure high quality knowledge extraction. Besides, knowing the semantic roles of arguments can improve the effectiveness of question answering.

In this work, we develop a *high precision knowledge extraction* pipeline tailored to the financial news domain by combining SRL information extraction for verb predicates with *typed* patterns for noun mediated relations. This pipeline filters noisy predicate-argument structures via a dictionary of semantically and structurally constrained *sense-disambiguated* financial predicates. In order to maximize the utility of the extractions for downstream tasks, our pipeline produces *compact* extractions via *dictionary-guided* minimization of overly-specific arguments. These extractions are scored using a binary classifier, with the score reflecting our confidence in the extracted fact. We perform a lossless decomposition of the n-ary relations extracted, to construct the KG. While some components of the the pipeline are customized to the financial domain, we believe with small tweaks it can be easily adapted to other domains.

Compared with (Benetka et al., 2017), the most closely related work, our pipeline extracts over 342,000 n-ary facts and covers more types of financial predicates – a total of 87 as opposed to 50 in (Benetka et al., 2017). Furthermore, our pipeline produces high precision extractions, specifically 78% at the top-100 extractions, as opposed to 34% of the pipeline from (Benetka et al., 2017).

In summary, our main contributions are as follows. We design a high precision knowledge extraction pipeline tailored to the financial news domain. Our pipeline combines SRL and pattern based information extraction to extract domain targeted noun/verb-mediated relations. We develop a Conditional Random Field (CRF) model that identifies and removes sequences of noisy text commonly found in financial news articles. To further improve precision, we build a dictionary of semantically and structurally constrained sense-disambiguated financial predicates to filter out noisy extractions produced by SRL. The ∼380,000 triples we extracted are stored in a KG which can be readily queried. We also conduct ablation studies to examine the effect of the different components of the pipeline on a number of performance metrics.

2 Related Work

Cross-domain KGs such as DBpedia, Freebase, NELL, and BabelNet (Navigli and Ponzetto, 2012) contain encyclopedic knowledge covering real world entities across different domains (e.g., people, organizations, and geography). They were either manually curated (e.g., Freebase) or automatically created, from semi-structured textual sources such as Wikipedia infoboxes (e.g., DBpedia), or unstructured text on the web (e.g., NELL). A number of efforts to create *domain targeted* KGs followed. The Aristo Tuple KB (Mishra et al., 2017) extracted 294,000 high-precision SPO triples using a KG extraction pipeline targeted towards elementary science topics from domain relevant sentences found on the web. In (Wang et al., 2018), the authors developed a framework (CPIE) which extracted relational tuples between 3 fixed types of biomedical entities from PubMed paper abstracts. In the financial domain, Crunchbase curated

a KB covering over 100,000 companies, investors, acquisitions and funding rounds, while (Benetka et al., 2017) extracted quintuples of monetary transactions covering 316 economic events.

NELL used a predefined ontology of categories and relations and a few seed examples that are used for semi-supervised bootstrap learning of semantic categories of entities and the relations that exist between them. This bootstrapping approach reduces the human labor required to label training data while taking advantage of the huge amount of unlabeled data available. While manually defining an ontology leads to high precision extractions, it requires domain experts and, in an open domain, it is not feasible to define a complete ontology. OpenIE aims to overcome these limitations by extracting all relational phrases in raw surface form in a single pass over the corpus. 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 (Benetka et al., 2017) 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)[2]. 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 (Miller, 1995). 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 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.

3 The Knowledge Extraction Pipeline

In the following subsections, we describe each component of our KE pipeline (see Fig. 1). The pipeline operates at the sentence level and starts with cleaning the news articles by identifying and removing noisy text spans, then performs linguistic annotations, i.e., resolves co-references and identifies named entities. Predicate argument structures are then extracted by the SRL component and passed to the financial predicate dictionary which we built to filter out noisy extractions by the SRL. 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. Finally, we score the predicate argument structures to reflect our confidence in their precision and conciseness. The input to the pipeline consists of two components:

(i) *Financial news corpus*: US Financial news[3] dataset containing ∼ 306k news articles collected from Bloomberg.com, CNBC.com, reuters.com, and wsj.com between January and May 2018;

(ii) *Financial Times Lexicon*[4]: this lexicon includes thousands of financial words and phrases selected by Financial Times editors (e.g., *capital ratio*, *corporate bond*, and *free market*). We use this lexicon to identify and minimize *overly specific* arguments as described in the minimization stage.

Text Pre-processing & Cleaning (CL). We start with standard NLP cleaning by removing brackets, parentheses, quotes and other punctuation marks. A more challenging and important cleaning task that is unique to our problem is to identify and remove *noisy text spans* present in the article, such as the publication date, the time the article was last updated, reporters' names, and/or reading time. This information is usually embedded within the article lead and is not separated from the content by any

[2] https://catalog.ldc.upenn.edu/LDC2008T19
[3] https://www.kaggle.com/jeet2016/us-financial-news-articles
[4] https://markets.ft.com/glossary/searchLetter.asp?letter=A

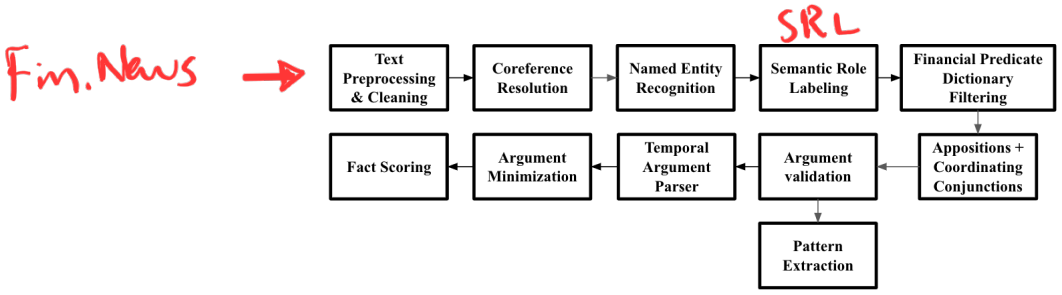


Figure 1: The financial knowledge extraction pipeline.

January 16, 2018 / 5:16 PM / Updated 8 hours ago
Health-conscious Nestle sells U.S. candy to Ferrero for $2.8 billion Martinne Geller, Francesca Landini
5 Min Read.

Figure 2: Noise in a news article lead. The colors highlight the noise boundaries.

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

Figure 3: Entry for *acquire.01* in the financial predicate dictionary; "r" is *required* while "o" is *optional*.

particular separating tokens (see Fig. 2 for an example). The variety in which noise of this kind can manifest in text limits the feasibility of using regular expressions to capture and eliminate such noisy text spans. Additionally, this type of noise can appear *anywhere* in the article and is not limited to its lead.

We cast the task of identifying noisy text spans as a sequence labeling problem where tokens in a sentence are assigned a sequence of labels. We manually annotated a dataset of 779 sentences containing 37% noise and trained a Conditional Random Field (CRF) (Lafferty et al., 2001) to label sequences of tokens using token features that combine information from the surrounding tokens and their part-of-speech tags. Specifically, for each token at position t, we extract unigrams and POS tags between positions $t-2$ and $t+2$ and use combinations of these features to describe the token.

Once the text is cleaned, we then resolve references to the same entity using the co-reference resolution system (COREF) integrated in the SpaCy NLP library (Honnibal and Montani, 2017). We identify named entities using the AllenNLP named entity recognition (NER) system (Gardner et al., 2017).

Semantic Role Labeling (SRL). We extract semantic relationships between entities using Semantic Role Labeling. As described in Section 1, information about the predicate sense and semantic roles of arguments in a relation are not captured by traditional OpenIE extractors, making them less useful in domain specific IE. Consider the sentence: *"Whole Foods was acquired by Amazon in 2017 for $13.7 Billion"*. OpenIE extracts (*Whole Foods; was acquired; by Amazon; in 2017; for $13.7 Billion*). While the OpenIE extraction is accurate, it is not useful for answering queries since it is unclear which entity acquired the other entity, or which argument is the price or the date. SRL, on the other hand, extracts **acquire.01**(*agent: Amazon, thing acquired: Whole Foods, price paid: $13.7 Billion, Temporal argument: 2017*). SRL not only identifies the correct *sense* of the predicate as *acquire.01* but also identifies the role of each argument. Correctly identifying the sense of the predicate and thematic roles of its arguments helps us impose *structural* and *semantic* restrictions to improve the precision. Concretely, the predicate *acquire.01* (*01* is the predicate sense meaning *get*) must have at least two arguments, one with the role: *entity acquiring something* and the other with the role: *thing acquired*. Further, in the financial domain, we would like to enforce that both arguments have type *ORG*, i.e., an organization. An optional argument is the *the price paid* whose type should be *MONEY*. We use LUND-SRL (Johansson and Nugues, 2008) for extracting and labeling predicate-argument structures. The output from LUND-SRL includes the lemmas, POS tags, and the dependency relations among all tokens in the sentence.

Financial Predicate Dictionary Filtering (FPDF). 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 (Baker et al., 1998) 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 (Kingsbury and Palmer, 2002). Fig. 3 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.Many of the SRL extracted relations contain temporal arguments *AM-TMP* such as *"today"*, *"last year"*, or *"3 months ago"*. We pass these arguments to a date parser library[5] that parses localized dates into a standard date format relative to the publication date of the article. 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*).

Appositions (APPOS). We produce implicit extractions from *appositions*. Consider the sentence: *"Dubai-based port operator DP World announced plans to transport cargo"*. The relation: **isA**(DP World, Dubai-based port operator), is extracted by following the *APPO* dependency relation between *"operator"* and *"World"*.

Coordinating Conjunctions (CC). We process *coordinating conjunctions* (CC) which join similar syntactic units (i.e., *conjoints*) into larger groups by means of CC such as *and/or*. Consider the sentence: *"HFF , HFF Securities L.P. and HFF Securities Limited are owned by HFF Inc."*. The argument *"HFF, HFF Securities L.P. and HFF Securities Limited"* has three conjoints and is considered overly specific. Extracting conjoints produces three relations with simpler arguments instead of one long argument. We find the CC using the dependency relations *coord* and *conj*.

Argument Validation (AV). Not only does the financial dictionary filtering step help ensure that the extractions are financially relevant, but its semantic and structural constraints also help eliminate predicate-argument structures that are incorrectly labeled by the SRL system. E.g., SRL correctly identifies the labels of (*Amazon*) and (*Whole Foods*) in the sentence *"Whole Foods was acquired by Amazon in 2017 for $13.7 Billion"*. Both of the arguments satisfy the entity type constraint *ORG*. Similarly, the arguments (*$13.7 billion*), (*2017*) are correctly identified and both have the correct types, i.e., *MONEY* and *DATE*, respectively. Consequently, all arguments pass the argument validation step. By contrast, consider the sentence *"Israeli-Palestinian relations sank to a low..."*. The SRL extracts **sink.01**:(*thing sinking*: Israeli-Palestinian relations, *end point, destination*: to a low). However, in the financial predicate dictionary, the *end point, destination* of the predicate *sink.01* must of one of the types (*MONEY*, *QUANTITY*, *CARDINAL* or *PERCENT*). As a result, this extraction is rightfully filtered out by this step.

Pattern Extraction (PTRN). In addition to the verb mediated relations extracted via SRL, we extract noun mediated relations via a pattern based extractor. The patterns are similar to those in the part-of-speech and noun chunks extractor (Pal and others, 2016), except that we add entity type constraints to the patterns, i.e., *ORG* and *PER*. This 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. For the patterns that do contain demonyms, we use the demonym-location table from (Pal and others, 2016). Overall, we extract 11 pattern types. Due to space limitations, we show the three most common pattern types in the corpus in Table 1. We create a string that replaces tokens with POS tags, entity types or demonyms, then check if it matches one of the patterns types.

Argument Minimization (MIN). In addition to processing coordinating conjunctions, we minimize

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

Pattern type	Pattern instance	Predicate(arg1, arg2)
Demonym-ORG CRN	German drugmaker Merck	**DrugMakerFrom**(Merck, Germany)
Demonym-PER CRN	French President Macron	**PresidentOf**(Macron,France)
ORG-PER CRN	Apple founder Steve Jobs	**FounderOf**(Steve Jobs, Apple)

Table 1: Pattern types, instances and extracted tuples.

arguments even further by identifying and dropping additional tokens that are considered overly specific. For this, we drop tokens that are considered *safe* to drop such as determiners, possessives, and adjectives modifying named entity PER (except demonyms). We then perform a *dictionary-guided* minimization of the noun phrase pattern $[adverbial|adjective]^+ Noun^+$ similar to the *dictionary mode* of (Gashteovski et al., 2017), except we use the Financial Times lexicon in place of the dictionary of frequent subjects, relations and arguments found in their corpus. This ensures that we do not drop tokens that are meaningful and important in the financial context. E.g., consider the sentence *"Mikros Systems Corporation , an advanced technology company, announced..."*. We extract **isA**(Mikros Systems Corporation, an advanced technology company). We then drop the determiner *"an"*, and proceed with enumerating sub-sequences of the noun phrase pattern instance *advanced technology company* and query its sub-sequences against the financial lexicon. Since "*advanced company*" is not found in the lexicon, we drop *advanced* from the argument.

Fact Scoring (SCORE). We score the predicate argument structures to reflect our confidence, by training a binary logistic regression classifier using over 1400 SRL extractions which we manually labeled. Extractions are considered valid if they are both *precise* and *concise*, i.e., explain only one proposition. We identified a collection of features that are powerful predictors of validity. The features include the presence of a coordinating conjunction, apposition, or verb, unresolved temporal arguments, pronouns, determiners, bad characters, and the predicate and named entities in the argument. We classify each valid argument of the extracted fact and take the minimum over all argument scores as the overall confidence score of the fact. Using the minimum aggregate function promotes the most precise extractions.

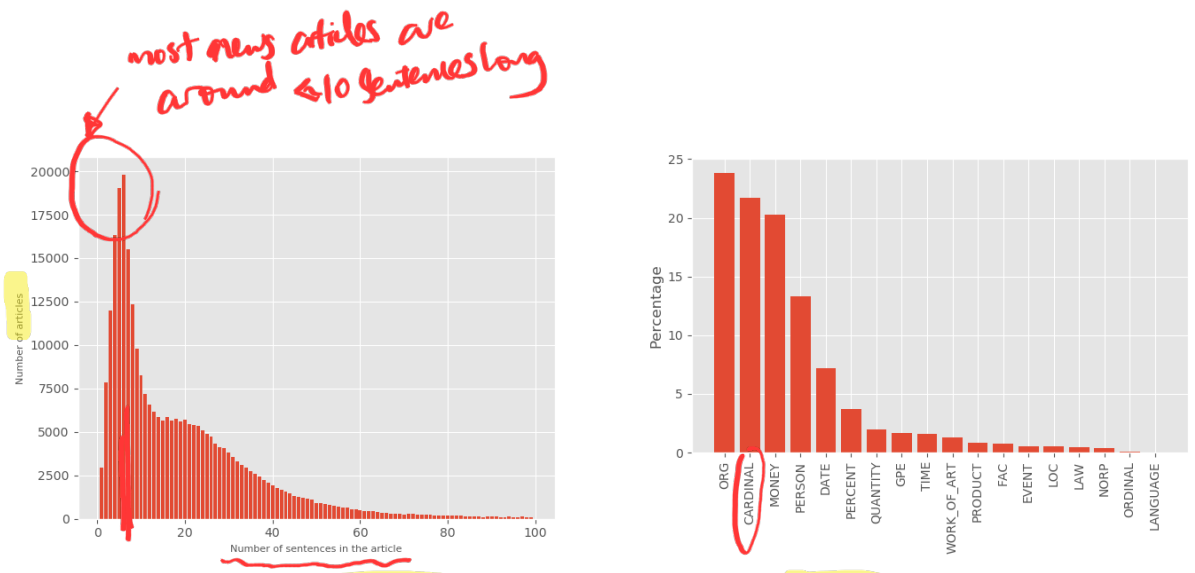
4 Evaluation

We run our KE pipeline on the US Financial News corpus. Fig. 4(a) shows the distribution of news article length in the corpus, measured by the number of sentences. For ease of exposition, the figure is trimmed by eliminating the distribution's long tail. Observe that most of the articles have fewer than 20 sentences and articles with more than 60 sentences in length are very rare. Fig. 4(b) shows the distribution of named entity types in the corpus. As expected in a financial news corpus, the top 4 entity types, *ORG*, *CARDINAL*, *MONEY*, and *PERSON* account for almost 80% of the unique named entities in the corpus. We report the results and compare with the work in (Benetka et al., 2017) which extracts *(subject, predicate, object, monetary value, date)* quintuples from the New York Times Annotated Corpus (NYTC)[6]. Further, we compare the functionalities of our pipeline against (Benetka et al., 2017) in Table 3. Finally, we illustrate a small subgraph that answers a query posed to the large extracted KG.

Extraction statistics. To demonstrate the effectiveness of the pipeline, we report in Table 2, a number of extraction statistics resulting from the processing of 288,118 articles. A total of 342,181 tuples were extracted by the SRL, pattern and apposition modules from 201,731 sentences, constituting 5.2% of all sentences in the corpus with average arity of 2.27. On the other hand, the pipeline in (Benetka et al., 2017) extracted only 496 quintuples from 2.1M sentences (of which only 18.2% describe economic events) from 1.8M articles. We found that 94.7% of the predicate-argument structures that were eliminated did not pass the financial predicate filtering step. This indicates that the financial dictionary filtering likely had the greatest impact on the total number of facts extracted by the pipeline.[7] It also suggests that the vast majority of the sentences in the corpus do not contain financially relevant facts. Another 3.69% of the relations did not pass the syntactic requirements whereas 1.52% did not pass the argument validation step. This suggests that the semantic and structural constraints on the predicates do not play

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

[7]Since it was not feasible to find the ground truth facts in our corpus, we could not measure the recall.



(a) Article length distribution

(b) Named entity type distribution

Figure 4

# (%) parsed temporal arguments	51505 (78.5%)
# pattern extracted facts	10828
# (%) *distinct* pattern extracted facts	4454 (1.3%)
# (%) SRL extracted facts	161983 (47.33%)
# (%) appositions	175744 (51.35%)
Average argument length in tokens	3.68 tokens
# tokens dropped by MIN stage (safe, dictionary)	427337(53.2%, 46.8%)

Table 2: Statistics and results of running our extraction pipeline on the corpus.

	Our Pipeline	(Benetka et al., 2017)
CL	✓	✗
COREF	✓	✗
NER	✓	monetary value recognition
APPOS	✓	✗
CC	✓	✗
FPDF	87 predicates with semantic and structural constraints	50 predicates, two types of econ. events
PTRN	✓	✗
MIN	✓	✗
SCORE	✓	✓

Table 3: Comparison between the functionalities of our pipeline vs (Benetka et al., 2017).

a major role in filtering candidate predicate argument structures, hence relaxing these constraints would not substantially increase the number of extractions. More than half of the facts were implicitly extracted via appositions. The SRL module extracted over 161,000 predicate argument structures contributing to 47.83% of the facts, whereas ∼1.3% were noun mediated relations extracted via patterns. It is important to note that of the ∼11,000 pattern extracted facts, only 4454 are distinct.

The minimization module responsible for condensing overly specific arguments dropped over 427,000 tokens. The majority of the tokens dropped this way were due to safe minimizations, i.e. determiners or possessives. The rest of the tokens were dropped by the dictionary-guided minimization. The top bigram lexicon hits were *quarterly dividend* (queried 372 times), followed by *common stock* and *net income*, and subsequently the tokens of these bigrams were marked as stable. The adjectives in these bigrams, i.e., *quarterly*, *net*, and *common* are critical in financial context. Thus, simply dropping adjectives would result in the loss of important information, and this is avoided by querying against the financial lexicon. This emphasizes the importance of the financial lexicon in preserving tokens that are important in the financial context while minimizing overly specific arguments.

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 the top 50, top 100, and top 150 extractions, i.e., Precision@50, Precision@100, and Precision@150 is 78%, 78%, and 79.33% respectively. The pipeline in (Benetka et al., 2017) has a much lower precision of 34% (at a recall of 20%).

KG Statistics. To build the KG, we break down each n-ary relation into binary relations (triples) identified by *predicate.sense-id* (e.g., *announce.01-0.8926653297234465*). Fig. 5(a) 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), and **acquired_in_0**: (*agent*: Merck, *AM-TMP*: 04/04/2015), where the suffix *0* is the relation *id*. The ID ensures lossless decomposition of an n-ary relation and helps to identify the different arguments. The resulting KG has 248,923 nodes, 474,837 edges (which becomes 380,079 edges after eliminating redundancies), and 31,144 weakly connected components (WCC), i.e. subgraphs where each pair of nodes is connected by a path, ignoring edge directions. The diameter, i.e., the longest distance between a pair of vertices, of the largest WCC is 19. The KG has an average degree of 1.5269.

Predicate Distribution. Fig. 5(b) shows the distribution of the top 10 financial predicates. The predicate *announce.01*, which describes the semantic frame "*Statement*" makes up 17% of the total SRL facts. Out of the top 20 predicates extracted, four: *increase.01, rise.01, fall.01*, and *decrease.01* describe the semantic frame "*Change position on scale*" which is usually associated with stocks reporting. This semantic frame, among others including *issue.01, launch.01* and *name.01*, were not captured in the ontology of (Benetka et al., 2017), which was limited to just two types of economic events.

Pattern Distribution. The most common pattern, i.e. the pattern that extracted the most facts, is *Demonym-ORG CRN* (an instance of this pattern is the fact: **Healthcare_group**(Sanofi, France) which was extracted 23 times), followed by *Demonym-PER CRN* (e.g., **President**(Donald Trump, United-States),[8] extracted 1026 times), and *ORG-PER CRN* (e.g., **Secretary**(Steven Mnuchin, Treasury), extracted 81 times). These three pattern types account for over 90% of the pattern extracted facts. Fig. 5(c) shows the top 10 relational nouns extracted via patterns.

Illustrating the KG. The entire knowledge graph is huge, thus for the purpose of illustration, we show a small subgraph of the KG that we extracted in Fig. 5(a). Consider the query "Which drugmakers were acquired by a German drugmaker?" (Query (1), Section 1) posed on the entire KG. The subgraph in Fig. 5(a) presents a subset of the answers to this query in the form of a graph. It says that `Merck` acquired both `Cubist Pharmaceuticals` and `Medco` along with details about date of acquisition and amount paid, where available.

5 Ablation Studies

We conduct ablation studies on a random sample of 1000 articles to gauge the effect of each stage of the pipeline on the overall performance. The results are reported in Table 4. Bolded cells in each column capture the most significant impact of turning off the corresponding module. Turning off the cleaning module (column **CL**) results in including noisy text spans and the overall #extractions drops (48.5% drop). More importantly, the precision in the top 50 and top 100 extractions drops significantly – by 12% and 4% respectively. Turning off the co-reference stage (column **COREF**) results in shorter sentences passing the sentence length filtering stage, yielding more sentences. The precision@50 increases by 4%. However, the total number of facts drops by 4.2% as significantly fewer SRL facts are extracted (since references are not resolved). Turning off the financial predicate filtering stage (**FPDF**) results in 18.2% increase in the total number of facts. However, that comes at a price of 68% loss in precision@50. Turning off the coordinating conjunctions stage (**CC**) results in a 2% increase in precision@50 for the price of a 16.5% drop in the total number of facts. It also results in a 4.2% increase in the average argument length. This demonstrates the effectiveness of **CC** in minimizing overly specific arguments. Turning off appositions extraction (**APPOS**) yields a 2% (resp. 4%) increase in precision@50 (resp. precision@100), at the expense of a 53% drop in the overall number of extracted facts. Turning off the minimization module (**MIN**) results in 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 owing to the financial lexicon guided minimization. Compared to the full pipeline (**NONE**), turning off any single module does not prune away the financial facts, except turning off **FPDF** shrinks the financial facts to a mere 14.1%, attesting to the crucial role this module plays.

[8] The most frequent pattern-extracted fact.

Metric	NONE	CL	COREF	FPDF	CC	APPOS	MIN
# sentences processed	13101	6521	13144	13101	13101	13101	13101
% sent. with facts	5.51%	5.56%	5.24%	58.7%	5.51%	**3.43%**	5.51%
# pattern/SRL/Appos.	24/590/665	**20/305/333**	**24/531/670**	611/18839/5118	**24/493/550**	**15/585/0**	31/590/665
Avg. argument length	3.80	3.83	3.75	3.56	**3.96**	**5.10**	**4.30**
P@50(P@100)%	82(79)%	**70(75)%**	**86(80)%**	**14(32)%**	84(80)%	84(83)%	80(77)%
% financial facts	100%	100%	100%	**14.1%**	100%	100%	100%

Table 4: Ablation studies results. *NONE* corresponds to running the full pipeline.

6 Discussion and Future Work

Our pipeline was effective in extracting predicate-argument structures from sentences with long range dependencies. E.g., from the sentence[9] "***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.***", our pipeline extracted the fact **secure.01**(*entity acquiring something*: 'Orb Energy', *thing acquired*: '$10 million', *source*: 'rooftop projects', *AM-TMP*: '02/14/2017'). 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. Furthermore, the pipeline successfully identifies and classifies the roles of different arguments to a fine granularity. E.g., from the 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% .*", we extracted **increase.01**(*thing increasing*: 'Parke Bancorp 's loans', *start point*: '$852.0 million', *end point*: '$1.01 billion', AM-TMP: '12/31/2017'). Such granularity is useful in answering complex queries such as *"find a contiguous sequence of stock prices that are all increasing (or all decreasing)"*. The pipeline successfully classifies less common predicate senses in difficult contexts: e.g., *settle.02* (meaning *resolve*) versus *settle.01* (meaning decide), and *cut.02* (meaning *reduce*) versus *cut.01* (meaning *slice*). Correct classification of predicate sense is critical for assigning correct semantic roles to the arguments. Our pipeline extracted ∼342,000 n-ary facts from only 5.2% of the sentences. One way to extract more facts is by expanding the financial predicate dictionary, while adding semantic and structural constraints on the new predicates to improve precision.

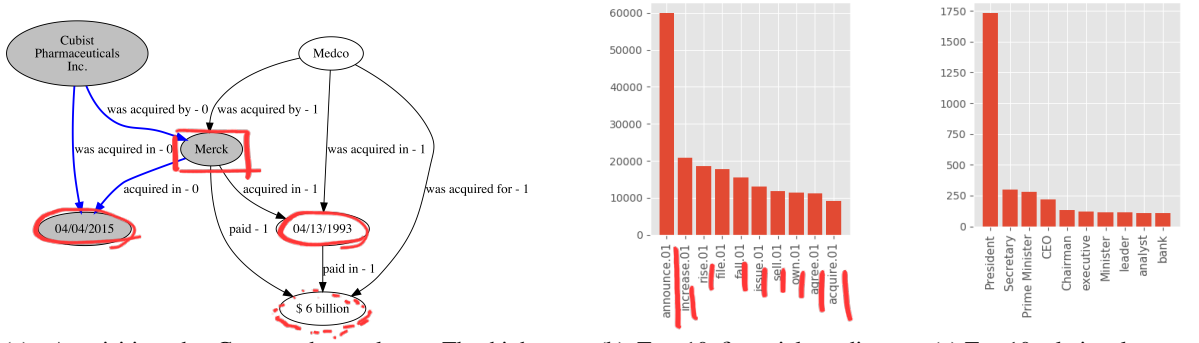
The ablation studies show that the financial predicate filtering was the most important factor for precision. Furthermore, the cleaning stage presents a significant trade off in precision and total number of extractions. Depending on the downstream applications, we may want to favor precision over number of extractions or vice versa. The study also 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 sacrificing the precision.

The ablation study shows the importance of the minimization stage in decreasing the average argument length. We would like to examine expanding the minimization beyond the safe and dictionary minimization of adverbial patterns. Prepositional phrases, although good candidates for minimization, are equally challenging. It is also common to see arguments such as "*more than 3%*", "*as much as 3 percent*", "*almost 3%*" in reporting stocks. We would like to *canonicalize* these arguments into $> 3\%$ or bin them into numeric ranges in order to maximize their utility in downstream applications. In future work, we would like to examine the *transferability* of the pipeline to other domains. The following adjustments will be needed: (1) a domain targeted dictionary for filtering candidate predicate argument structures; (2) a domain targeted lexicon for dictionary minimization; and (3) supervised models for cleaning and fact scoring trained on datasets from the target domain. We would also like to explore methods to build a domain targeted dictionary using corpus level statistics and/or learn models that incorporate consistency constraints and automatically identify fact relevance using triple relevance features.

7 Conclusions

We developed a high precision pipeline for knowledge extraction from a financial news corpus that produced over ∼342,000 n-ary facts at 78% precision by employing multiple information extraction

[9]Emphasis added.



(a) Acquisitions by German drugmakers. The highlighted nodes and edges represent one relation.

(b) Top 10 financial predicates.

(c) Top 10 relational nouns.

Figure 5

techniques. We built a financial predicate dictionary that places structural and semantic constraints on arguments to produce high quality extractions. To enhance the utility of the extractions, we minimized overly specific arguments by processing coordinating conjunctions and appositions, and employed a financial lexicon to minimize adverbial nouns. We evaluated the pipeline and the resulting KG on a number of metrics and conducted ablation studies to examine the effect of different modules of the pipeline on these metrics. This study 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.

References

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

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

Jan R Benetka, Krisztian Balog, and Kjetil Norvag. 2017. 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.

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

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

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

Kiril Gashteovski, Rainer Gemulla, and Luciano del Corro. 2017. 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. Association for Computational Linguistics.

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

Shaoxiong Ji, Shirui Pan, Erik Cambria, Pekka Marttinen, and Philip S Yu. 2020. A survey on knowledge graphs: Representation, acquisition and applications. *arXiv preprint arXiv:2002.00388*.

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

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

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

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

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

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

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

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Natasha Noy, Yuqing Gao, Anshu Jain, Anant Narayanan, Alan Patterson, and Jamie Taylor. 2019. Industry-scale knowledge graphs: Lessons and challenges. *Communications of the ACM*, 62 (8):36–43.

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

Edward Shortliffe. 2012. *Computer-based medical consultations: MYCIN*, volume 2. Elsevier.

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

Xuan Wang, Yu Zhang, Qi Li, Yinyin Chen, and Jiawei Han. 2018. 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.