machine learning & knowledge extraction

Article

Knowledge Graph Extraction of Business Interactions from News Text for Business Networking Analysis

Didier Gohourou * and Kazuhiro Kuwabara

Graduate School of Information Science and Engineering, Ritsumeikan University, Kusatsu 525-8577, Shiga, Japan
* Correspondence: gr0259sh@ed.ritsumei.ac.jp

Abstract: Network representation of data is key to a variety of fields and their applications including trading and business. A major source of data that can be used to build insightful networks is the abundant amount of unstructured text data available through the web. The efforts to turn unstructured text data into a network have spawned different research endeavors, including the simplification of the process. This study presents the design and implementation of TraCER, a pipeline that turns unstructured text data into a graph, targeting the business networking domain. It describes the application of natural language processing techniques used to process the text, as well as the heuristics and learning algorithms that categorize the nodes and the links. The study also presents some simple yet efficient methods for the entity-linking and relation classification steps of the pipeline.

Keywords: knowledge extraction pipeline; graph dataset; business networking

check for updates

Citation: Gohourou, D.; Kuwabara, K. Knowledge Graph Extraction of Business Interactions from News Text for Business Networking Analysis. *Mach. Learn. Knowl. Extr.* **2024**, *6*, 126–142. https://doi.org/10.3390/make6010007

Academic Editor: Irena Spasić

Received: 23 October 2023
Revised: 27 December 2023
Accepted: 2 January 2024
Published: 7 January 2024

1. Introduction

The field of machine learning has seen significant advancements in algorithms for graph analysis, which have proven effective in tasks such as node classification, link prediction, and graph classification [1]. This has sparked a renewed interest in graph theory applications across diverse fields notably network science and knowledge graphs (KGs) [2], which are pivotal in areas ranging from molecular biology to social network analysis. The ubiquity of graph data structures and the topological advantage they present, as emphasized by Bronstein et al. [3], makes them suitable for deep-learning models that consider the geometry of data with irregular topologies.

Molecular biology, social networks, recommender systems, and question-answering are a few examples of the domains that benefit from deep-learning models trained on graph data structures. For instance, molecular biology leverages deep-learning models trained on graph data structures to enable analysis of protein–protein interaction networks [4]. In social networks, graph-based datasets facilitate community and anomaly detection [5,6]. Recommender systems harness graph-based models to improve personalized recommendations [7]. Additionally, question-answering benefits from the knowledge graph representations of data by capturing semantic relationships between entities [8]. Another domain that can greatly benefit from deep learning on graph-structured datasets is the study of business networking. Graph-structured datasets can improve business networking in many ways, including relationship analysis of connections between businesses to uncover potential opportunities for partnerships or collaborations and community detection to identify entities that share common interests. Business networking analyses are usually conducted from the perspective of an organization to understand the relationships with customers and suppliers. Leveraging graph-structured data will enable market-level business network analysis, encompassing cross-market business networking. This will provide business decision-makers with unprecedented insights.

Yet, a notable gap persists in the availability of graph datasets tailored for business networking analysis, particularly in benchmark datasets such as the Open Graph Benchmark [9]. Business news articles, press releases, and industry publications can be valuable for tracking business events, partnerships, mergers, and other networking activities. They provide timely and current information, which is crucial for understanding the latest developments in the business world and how they impact networking relationships. Most news articles are publicly accessible through the web, cover a diverse range of industries, and are expected to be unbiased. It is worthwhile to have tools that seamlessly generate graph-structured datasets from business news articles for the business networking domain to take advantage of advances in the graph-learning field.

This study introduces TraCER (Trading Content Extraction and Representation), a streamlined and efficient pipeline designed to extract and represent content from unstructured text data. The primary objective of TraCER is to generate a graph representation dataset tailored for the analysis of business networking. Constructing networks from unstructured textual information presents unique challenges, with the foremost being the identification of entity references and their interconnected relationships. While the literature contains various studies and tools dedicated to distinct aspects of automatic knowledge graph construction from text, such as named entity recognition [10] and relation extraction [11], there is a scarcity of integrated approaches that seamlessly transition from textual data to graph creation. Moreover, to the best of our knowledge, no such automated pipeline has been specifically designed for the domain of business networking.

This study's contribution, therefore, seeks to fill this gap by addressing the absence of domain-specific graph datasets for business networking. We achieve this by providing an integrated toolchain that enables the generation of business networking graph datasets from business news sources. The tools comprising TraCER are openly accessible on the web (https://github.com/semlab/tracer and https://github.com/semlab/triplex, accessed on 27 December 2023).

2. Related Work

2.1. Building a Knowledge Graph from Text

Transforming textual data into network representations is commonly a challenging endeavor. Typically, networks are constructed using data sourced from knowledge bases. The creation of these knowledge bases often involves either an extensive and meticulous process of data collection and curation, carried out by experts in the domain, or through an automated compilation of diverse and external structured data sources. Consequently, the development of a network dataset tends to be either a labour and time-intensive task or requires a sophisticated combination of tools to extract and combine the pre-existing structured data.

The representation of data as a graph was initially carried out manually. As such, the Zachary karate club [12] was put together as a network representation. It is a small dataset of 34 nodes, which is frequently used as a benchmark dataset, including in contemporary studies [13,14]. Considerable human labor went into building knowledge graphs at scale. Some examples include WordNet, a lexical database of semantic relations between words [15], and Cyc, which attempt to model basic concepts and rules of the world to be used as common sense knowledge by semantic reasoners [16]. The manual approach quickly turned out to be impractical at scale. This led researchers to introduce automatism in building knowledge graphs using structured knowledge bases. An example is CiteSeer [17], an index for academic literature that is autonomously built from citation sections of academic documents. As information retrieval methodologies improved, and catalogs of human knowledge became available on the web; large-scale KGs were built from information extracted from semi-structured information catalogs, notably Wikipedia. Those KGs include DBPedia and YAGO. In addition, as natural language processing methods are used for information extraction from natural text, the automation in the construction of the KG increased as well. The automatic construction of knowledge graphs involves many steps that are automated in isolation. Zhong et al. [18] surveyed procedures used to automate different steps of knowledge graph construction, in which we can notice that the automation of each step is an active domain of research. Fewer works offer an integrated approach that describes how to go from text to a KG. Kertkeidkachorn et al. propose T2KG, an automatic knowledge graph creation framework from natural language text [19]. T2KG extracts triples from texts then maps entities and predicates to an existing KG such as DBPedia. T2KG then provides a mechanism to build an open domain KG from texts.

As an alternative to open domain KGs, domain-specific KGs offer many advantages, including noise reduction, improved relevance, better precision and accuracy, and seamless integration with domain-specific applications when the area of interest is targeted. Domain-specific KG construction methods have been developed for e-commerce and applied to build a question-answering task for a product compatibility recommender system [20]. This was achieved using questions-and-answers text data related to products from an e-commerce website to generate triples. A KG is constructed from this set of triples according to a defined ontology. The KG is stored as RDF triples and used to automate responses for an e-commerce question-and-answering system, without the direct assistance of human attendants. Another account of a methodology that turns text into a domain-specific KG has been demonstrated over a variety of unstructured text to build a question-answering mechanism for movies [21]. The study uses an open information extraction implementation [22] to generate a list of entity–relation triples. Entities and relations from the extracted triples are encoded using the BERT language model [23] before being used to build a knowledge graph. The obtained KG is used for question-answering, where the author's experiment demonstrates the efficiency of their multi-hop KG traversal and retrieval mechanism. The study also emphasizes the ability of the presented mechanism to build a knowledge graph without alignment with an external knowledge base, unlike some other prevalent studies on knowledge graph construction approaches [24,25].

2.2. Business Network Knowledge Graph Construction and Analysis

Current literature on business networking analysis predominantly focuses on the perspective of individual companies understanding their relationships with their counterparts. One such study [26] describes the construction and exploration of and enterprise knowledge graph from a private structured database containing information on 40,000,000 companies. Their results allow the visualization of companies' interconnections, finding the real stakeholders in control of a company, discovering innovative companies that securities establishment would like to invest in, and understanding various types of relationships between companies including competition, patent transfer, investment, and acquisition. Another study [27] demonstrated how a graph-learning model can elicit cooperation and competition links between companies by embedding a graph build from structured data as well. They demonstrate the business value of their finding with a competition and cooperation analysis, showing how their results can be useful to a company's potential partners and competitors and also empowering analysts with insights to partners with the competitor of their competitors, citing the "enemy of my enemy is my friend" principle. We believe those studies will benefit from going beyond structured data. Hillebrand et al. [28] fine-tuned BERT to identify specific concepts such as key performance indicators (kpi), current year monetary value (cy), and davon, that are defined by the authors. In contrast to building a graph, their focus is on analyzing business documents, which is our domain of study, and they demonstrate an approach in handling specific entities within unstructured text documents.

Noticing insufficient attention given to the macroscopic analysis of business networks at a market level, we previously demonstrated the preliminary steps for building a graph for business networking analysis from unstructured text [29]. We also showed that using the obtained graph we can make use of machine learning for graph models to classify the nodes, hence making some inferences about the type of node extracted from the text dataset. In this study, we present the design and implementation of a systematic pipeline for the automatic construction of a KG from text. We built a KG to understand interconnections between organizations, people, places, and products. We thus create a tool that generates graph data sources for providing insights into business networking-related tasks.

3. Materials and Methods

TraCER is a comprehensive toolchain designed as an integrated pipeline, facilitating the transformation of text into a knowledge graph. At its core, the process encompasses several critical subtasks, each contributing to the effective conversion of textual data into a graph format. The pipeline is initiated by a (i) preprocessing task that prepares the text as suitable content to be processed in subsequent stages. From the preprocessed text, (ii) word embeddings are computed, while (iii) relationship triples are extracted. The triple extraction step includes open information extraction (OpenIE), named entity recognition, and a filtering process for noise reduction. The extracted triples are then (iv) categorized. Lastly, (v) the graph is created from the extracted entities and categorized relationships. Figure 1 gives an overview of the methodology. Before describing the steps of our method below, we start by defining the scope of the information we extract to form our knowledge graph using a simple ontology.

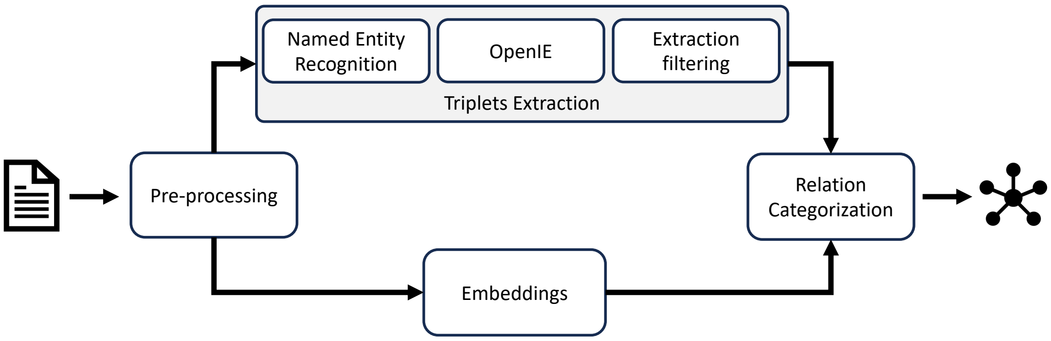

Figure 1. Overview of the proposed method.

3.1. Ontology

Hogan et al. [2] identify three ways of representing a knowledge graph. The first one is a directed edge-labeled graph that represents a set of nodes and a set of directed labeled edges between those nodes. The nodes can represent any concept. The second is an heterogeneous graphs where each node and edge is assigned a type. The third one is a property graph for additional flexibility to the heterogeneous graph. In this study, we opt for an heterogeneous graph. We determine the scope of our knowledge graph with the following ontology competency questions [30,31]:

- What is the nature of the relationship between Company A and Company B?
- Who produces Product X?
- What organizations operate in City Y?
- Who are the people working with Company C?
- Who buys Product Y?

Then, we derive an ontology from schema.org (https://schema.org, accessed on 27 December 2023) with entity types that include organization, person, place, and product. The entity type Organization represents such concepts including the company and business. The entity type Person represents the concept of people. The entity type Place represents the concept of a geographical place. The entity type Product represents products. The relationships are labeled according to the types of entities they involve. When the type of entity involved in a relationship is an organization or person the relationship type is either in competition or collaboration with, it is denoted as “collaborates with” or “competes against”, respectively. When the relationship is between an organization and a place, the relationship type is “operates in”. For a relationship involving an organization and a product, possible relation types are “consumes” and “produces”. Table 1 summarizes the valid relationships with the type of entities as the source and destination, as well as their type of relationship. Figure 2 provides a visualization of a possible graph based on the defined ontology.

Table 1. Type of source and destination entities and their possible relationship types, derived from our ontology.

Source (Subject)	Destination (Object)	Relationship Type
Organization, Person	Organization, Person	collaborates, competes
Organization	Product	produces, consumes
Organization	Place	operates in

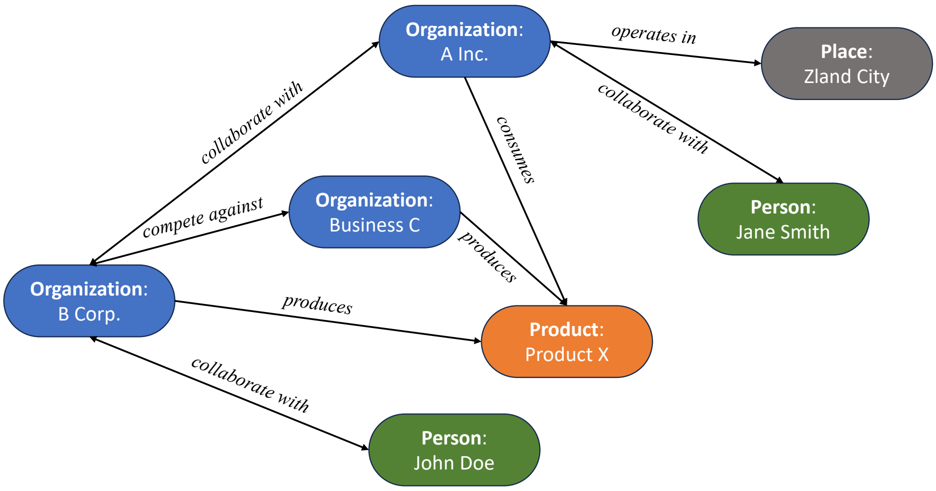

Figure 2. Visual of a possible extraction based on the defined ontology.

3.2. Preprocessing

Starting with text available on public news websites, the preprocessing phase aims at normalizing the text data. It consists of a preparatory phase that encompasses a series of operations. It includes the extraction of meaningful content by parsing raw text files notably in HTML format, the handling of intricacies such as duplicate entries and missing data special characters, as well as the elimination of HTML tags. Additional operations typically include tasks such as tokenization, lowercasing, removing punctuation, and stop word removal. Considering those preprocessing steps underpins the efficacy and precision of the natural language processing (NLP) models we will be using downstream in our task, facilitating the extraction of salient data and fostering a higher degree of computational efficiency.

3.3. Word Embedding

Language models [23,32,33] and, recently, large language models [34–36] have made significant advancements in NLP tasks due to their ability to model context and generate coherent text, but they come with their own set of challenges, such as high computational requirements, ethical concerns, and potential biases. At this iteration of our study, we opt for word embedding for the representation of the text we manipulate for three main reasons. The first reason is computational efficiency. In many real-world applications, computational resources are limited. Word embeddings are far more computationally efficient than large language models. Furthermore, training word embedding from the ground up is feasible using consumer-grade computers. The second reason is task-specific focus. Because our study is specific to the domain of business networking, we want our text representation to rely on a model trained on the very corpus from which we extract the graph. Fine-tuning a large language model is helpful for domain-specific tasks but at the cost of excessive computational demands and the risk of salient data being buried in a relatively large amount of generic text data used to pre-train the language model, in case of scarcity of the text data of interest. The final reason is that our method combines interpretable models in contrast to a large language model, which can reduce the stochasticity of the expected result. An additional reason is how word vectors from word embeddings are used in the relation categorization step of our method.

The skip-gram model from word2vec [37] is the one used to generate word vectors from the corpus obtained after preprocessing. Skip-gram builds word vectors by maximizing the log probability of a word appearing in a window context of a given center word.

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0} logp(w_{t+j}|w_t),$$

where T is the number of words in the vocabulary, i.e., the number of unique words in the corpus. $p(w_{t+j}|w_t)$ is defined using the softmax function:

$$p(w_O|w_I) = \frac{exp({v'_{w_O}}^{\top} v_{w_I})}{\sum_{w=1}^{W} exp({v'_{w_O}}^{\top} v_{w_I})},$$

where w_o is a context word, w_i is a center word, and W is the size of the vocabulary. v_{w_o} and v_{w_i} are the embedded numerical vector representations, respectively, for a context word and a target word. The final v_w vector representation of a given word w in the vocabulary is computed by pooling its vector representation as a context v_{w_o} word and its vector representation as a center word v_{w_i} [38].

3.4. Triplet Extraction

Our methodology employs open information extraction (OpenIE) for the automated identification and extraction of relationships, facts, and entities from unstructured text without relying on human intervention. OpenIE typically starts by segmenting sentences from the text. Then, it performs part-of-speech tagging and dependency parsing on each sentence to identify its grammatical structure, including the roles of words and their relationships within the sentence. To simplify the identification of relations in sentences, nominalization is applied to transform verbs into nouns. Candidate relations and their corresponding arguments (entities or noun phrases) are identified based on the grammatical structure of the sentences using patterns and heuristics. Relations are extracted as fact triples that consist of a subject, a relation, and an object.

We apply a specific OpenIE approach [39] that generates triples by first breaking a long sentence into a short coherent clause. This is achieved by modeling the task as a search problem with three types of actions: yield, recurse, and stop. Secondly, finding the maximally simple relation triple is warranted given each of these clauses. This is achieved by adopting a subset of natural logic semantics dictating context in which lexical items can be removed. Then, the short entailed sentences are segmented into conventional open information extraction (OpenIE) triples.

We added a filtering process on top of the triplet extraction to only keep triples of interest for our task. The filtering process consists of keeping triples where both the subject and the object contain named entities type of interest, as defined in the Section 3.1. Algorithm 1 provides pseudocode of the filtering process. *ExtractTriple* is a subroutine that extracts triples from a sentence using OpenIE. *FindNamedEntity* is a subroutine that identifies named entities within parts of the extracted triple such as the subject or the object.

Algorithm 1 Triplet Filtering

```
Require: $\mathcal{S}$ the set of sentences in the dataset
Require: $\mathcal{C}$ the set of named entity types of interest
  $T \leftarrow \emptyset$ /* Set of filtered triples */
  for all $s \in \mathcal{S}$ do
      $T_s \leftarrow$ ExtractTriples($s$)
      for all $t \in T_s$ do
          $s_e \leftarrow$ FindNamedEntity($t.subject$)
          $o_e \leftarrow$ FindNamedEntity($t.object$)
          if $\exists$ type($s_e$) $\in \mathcal{C}$ and $\exists$ type($o_e$) $\in \mathcal{C}$ then
              $t \cup T$
          end if
      end for
  end for
```

The named entity recognition (NER) process relies on the tokenization, part-of-speech tagging, and lemmatization of the text. The named entities are classified with a maximum entropy Markov model [40]. From most pre-trained NER models, we can recognize and filter entity types appearing in our ontology, namely organizations, people, and places. We customize the model to tag and identify another entity type of interest for our domain of study: products.

Furthermore, *type* returns the type of a given named entity. Considering $\mathcal{S}$ the set of sentences and T_{s_max} the largest set of triples extracted from a sentence, the proposed triple-filtering process exhibits a complexity of $O(|\mathcal{S}| \times |T_{s\_max}|)$, with $|\mathcal{S}|$ and $|T_{s\_max}|$ the cardinalities of $\mathcal{S}$ and T_{s_max}, respectively.

3.5. Reducing Duplicates

To mitigate the impact of noisy and duplicate data in our extraction process, we implement two key strategies: coreference resolution and a heuristic-based entity-linking process. Coreference resolution identifies different textual expressions referring to the same entity, enhancing our understanding of entity relationships. This results in a better understanding of the relationships between entities mentioned differently in the text. Examples of solved mentions are pronouns (e.g., "he", "she", "it") and generic entity coreferences (e.g., "the former"). For further redundancy reduction in identified relationships, we have developed a heuristic-based entity-linking process, detailed as follows:

Definition 1. *Let t be a triple extraction in $\mathcal{T}$ the set of extracted triples. Let e be a named entity with $type(e)$ the entity type of e, the object of t. e', the named entity in the subject of t, is a link of e if the text of the relation part of t is* 'be' *and $type(e) == type(e')$.*

This allows us to link two entities that belong to the same extraction triple for which the subject and object consist only of one named entity. The subject and object's named entities have the same type, and the predicate of the triple is the word "be". Illustrative examples are provided in Appendix B.

3.6. Relation Categorization

Upon completing the previous steps, our methodology yields a network representation where nodes represent named entities of interest, and links are labeled according to relationships from OpenIE-generated triples. To further refine our network, we categorize these links. Implementing this step is beneficial for downstream graph-related tasks such as link prediction which require knowledge of link types.

Link categorization is inherently a classification task, typically approached via supervised learning which necessitates labeled data. However, aiming to simplify the construction process of our study, we opted for an unsupervised learning approach, minimizing the need for extensive data labeling.

A vector produced using word-embedding algorithms such as Skip-gram [37] can be manipulated with a vector offset method to identify linguistic regularities in continuous space word representations [41]. Using algebraic operations on the vectors representing the words, analogies can be drawn. A frequently used example is $king - man + woman = queen$, meaning that the closest word vector to the one we obtain when subtracting the vector of the word *man* to the vector representing the word *king* then adding the vector *woman* is the vector that represents the word *queen*. We can deduce that $king - man \approx queen - woman$, assuming that the "$-$" operator acts as a pooling mechanism that gives a vector that represents the relationship between the word *king* and the word *man*, which should be close, i.e., approximately equal to the vector that represents the relation between the word *queen* and the word *woman*. Using this insight, we compute a vector representation for each relationship triple extracted using OpenIE by applying the vector offset method [41] to the vectors representing the named entity in the subject and the one representing the named entity in the object. Relationship vectors are then grouped according to the type of entities involved in their subject and object, as described in the Section 3.1. For each group, a clustering algorithm is used to assign the type of relationship.

Conforming to the ontology, and going through the described graph-building process, we ensure the construction of a domain specific knowledge graph for business networking and interaction, cleared of noisy entity types and irrelevant relationships. However, it is important to note that relationships not defined in the ontology are omitted from the graph. Further implications and limitations are described in Section 5.2.

3.7. Implementation Details

The TraCER pipeline is predominantly implemented using the Python programming language, except for the triple extraction. Preprocessed text datasets are formatted as comma-separated values (CSV) files, serving as input to the pipeline. The CSV format includes, in sequential order, the article's text, title, publication date, and topic. It is worth noting that only the article content is mandatory, while the other data fields remain optional. The embeddings are computed using the Gensim [42] Python library's Skip-gram implementation.

The triple extraction feature, named triplex, is implemented directly in Java, which is the programming language used by the Stanford CoreNLP library (https://stanfordnlp.github.io/CoreNLP, accessed on 27 December 2023), instead of using a Python wrapper, for performance reasons. We chose the Stanford CoreNLP library for its comprehensive set of features, which align well with our implementation's requirements [43]. Triplex is responsible for coreference resolution, open information extraction, and named entity recognition. Our setup includes tokenization, sentence splitting, part-of-speech tagging, lemmatization, named entity recognition, and open information extraction. The triple-filtering algorithm is implemented on top of the triple-extraction setup. We employed additional configurations for named entity recognition where necessary. Since we are considering the field of business networking, we are interested in entities that represent organizations, places, people, and products. The CoreNLP library inherently recognizes the first three entity types. To address products, we curate a specific set, which is then integrated into the named entity recognition system using the additional *TokensRegexNER* configuration mechanism within the library. The entity-linking mechanism takes as input the triples generated by triplex to identify linked entities and eliminate reflexive relations.

Following this step, the relationship categorization feature employs the outputs of the embedding stage and triplex to represent relations and categorize them as described in Section 3.6. In the process of extracting triples from the text, we assign relationship types to the extracted triples based on our predefined ontology. This assignment follows a structured approach: First, we group relationships according to the types of entities involved. Next, we represent these relationships as vectors using the vector offset method, which leverages the word vectors of the two entities linked by the relationship. Within our ontology, each group of relationship pairs typically falls into one of two categories: those with a single relationship type and those with two relationship types. For the former, we straightforwardly assign the sole relationship type. However, in the case of groups with two relationship types, we employ K-means clustering to categorize these relationships effectively. The resulting K-means clusters of vector relationships can then be labeled by human analysts for interpretability and context. We use the K-mean clustering algorithm implementation from the Scikit-learn Python package [44].

From the result, we build a graph using NetworkX 3.2 (https://networkx.org/, accessed on 27 December 2023), a Python 3 package for the creation and manipulation of complex networks. NetworkX was selected because of the rich features it offers to study the structure and dynamics of networks, and its network representations are also compatible with notable graph-learning packages, such as Deep Graph Library 1.1 (https://www.dgl.ai/ accessed on 27 December 2023), that we use down the line for predictive tasks.

4. Results

We applied our proposed methodology to a corpus related to business news from the web.

4.1. Dataset

The Reuters-21578 is a collection of documents that appeared on the Reuters newswire in 1987. The documents were assembled and indexed with categories by personnel from the Reuters Ltd. news company. The version of the dataset used in this study, which contains 21,578 documents, is a curated collection that was made available in 1996. This dataset was specifically built to serve as a resource for corpus-based research, in areas such as information retrieval and text categorization.

This dataset is suitable for our study because its size allows for relative rapid prototyping and the implementation of solutions for hypothesis testing. It offers various amounts of metadata related to the documents including dates and topics which can be leveraged in building insightful models. Moreover, the Reuters-21578 dataset is freely accessible throughout the web (https://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html, accessed on 27 December 2023), making it easy for third-party reproduction of the obtained results.

4.2. Graph Extraction

Our experimental dataset is provided as SGML (Standard Generalized Markup Language) files, which is a hierarchical tag structured markup language, similar to HTML. All the preprocessing steps described in Section 3.2 targeting HTML files had to be applied to the Reuters-21578 dataset to extract the article content as plain text. During preprocessing, special characters not pertinent to the newswire content were removed. Articles focused on earnings, which essentially consist of tables of numbers and articles with less than a hundred characters, were discarded, resulting in a dataset of 16,828 articles.

Given the small size of our experimental dataset, we selected the following key hyperparameters when computing the embeddings. The window size, which describes the range of context, was set at five words, and the vector size of the embeddings at 50 dimensions.

Using the triplex tool for triple extraction on the curated Reuters-21578 dataset yielded 13,842 triples. Table 2 provides a sample of identified named entities from the extracted triples.

We also applied our entity-linking heuristic to identify entity links from the triple. Despite being simple, the heuristic is quite effective at linking the entities to their different names including abbreviations and names in other languages. The triples extracted are relations that involve named entities of interest appearing both in their subject and object, with reduced noise, redundancies, and linked entities. Table 3 is a sample list of recognized entity links from the triples.

Table 2. Sample of identified named entities from the Reuters corpus and their associated type.

Entity	Entity Type
Youcef Yousfi	Person
Sonatrach	Organization
Cabot Corp	Organization
CBT	Organization
Frank Spadine	Person
Bankers Trust	Organization
Shearson	Organization
Tulis	Person
gas	Product

Table 3. Sample of identified entity links.

Entity	Linked Entity
National Association of Realtors	NAR
ChemLawn Corp	CHEM
Securities and Exchange Commission	SEC
Government Accounting Office	GAO
Swiss Mortgage Institute	Schweizerischer Hypothekarinstitute
General Petroleum and Mineral Organisation	Petromin
Cassa Di Risparmio Delle Provincie Lombarde	CARIPLO
Nippon Telegraph and Telephone Corp	NTT

After applying the entity-linking heuristic, we obtained 11,849 triples of interest. A sample of extracted triples is provided in Table 4. We evaluate the linking heuristic using an F-measure, the $F1$ score which is the harmonic mean of precision and recall, calculated using:

$$F1 = \frac{2 \times precision \times recall}{precision + recall},$$

where

$$precision = \frac{TP}{TP + FP},$$

and

$$recall = \frac{TP}{TP + FN},$$

with TP (True Positive) being the number of samples that are correctly identified as entity links, FP (False Positive) being the number of samples that are incorrectly identified as entity links, and FN (False Negative) being the number of samples that are entity links that were not identified by the heuristic. We measured an $F1$ score of 0.82, showing that the heuristic can successfully identify local entity links, even though there is room for improvement.

Table 4. Sample triple extractions from the Reuters corpus after running OpenIE with the triples filter.

Subject	Relation	Object
Youcef Yousfi	general of	Sonatrach
Cabot Corp	be	CBT
Frank Spadine	economist with	Bankers Trust
Shearson	have	Tulis
gas drilling	in decline be	Tulis

For the relation categorization step, the clustering accuracy was evaluated using the adjusted random score. From the Reuters dataset extractions, a K-mean clustering of relationships involving organizations and persons effectively put 86% of collaborative and competitive relationships in the same cluster. Relationships involving organizations and products were effectively clustered into production and consumption relationships with an 82% accuracy score using the same approach. Excluding relationships with one category, this gives an average accuracy of 84% on the Reuters dataset.

Putting the text dataset through the proposed pipeline, we can represent the data as an interconnection of nodes representing entities of interest and their categorized interconnection. Figure 3 is an overview of the resulting network. Appendix A provides labeled graphs extracted from sampled articles from the Reuters-21578 dataset for detailed visualization.

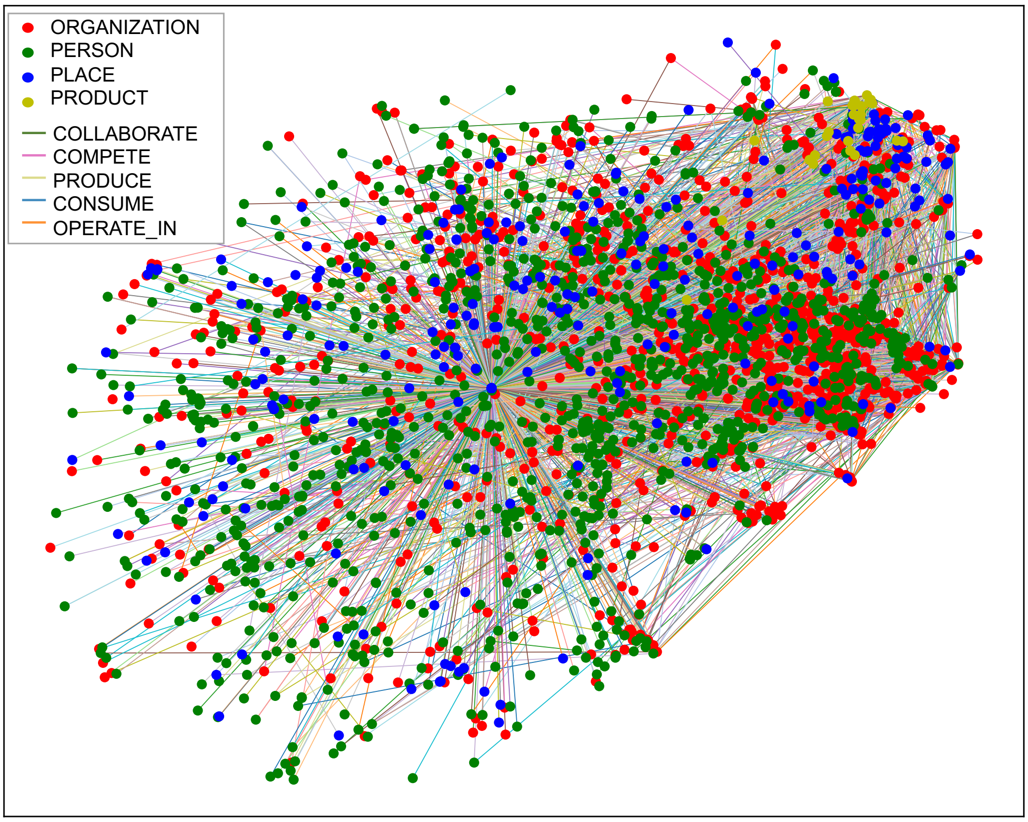

Figure 3. Visualization of the obtained network after running the pipeline over the entire preprocessed Reuters-21578 corpus. Entities' nodes are positioned according to the 2D projection of their vector representation. The node colors determine the type of entities and the link colors the group/class of the relation.

5. Discussion

Unlike existing text-to-knowledge graph approaches that use sentence stubs like verbs or noun phrases as link labels [21], our pipeline classifies relations within predefined categories, requiring minimal resources and no annotations. The categorized links present an opportunity for the link prediction algorithm to predict the nature of links between two entities in addition to predicting the existence of links.

Our experimental dataset is modest by contemporary standards. This can raise the question of the scalability of our methodology. While this is not addressed in this study, we leave it as a prospect for future work. One way to address the scalability is to apply our methodology to subsets of a large-scale text dataset and merge the resulting networks. For instance, Wu et al. [45] have demonstrated state-of-the-art methodologies in this area.

Additional preliminary steps were considered before building the network, including coreference resolution and entity linking as well as a reduction in noisy data such as duplicated entities in the graph by more than 70% compared to the graph generated from the previous implementation [29]. Appendix C provides an ablation study for a detailed analysis of the impact of key subtasks in reducing redundancies during the construction of the graph.

5.1. Use Cases

In our previous work [29], we showcased the embedding of the resultant graph using a random walk approach [13]. We then employed node property classification to recover missing node properties, such as entity types [29]. In this section, we extend our exploration of potential use cases for the generated graph, particularly within the realm of business networking analysis.

A natural application that emerges is graph neural-network-based link prediction, which is instrumental in uncovering previously undetected interactions between organizations or entities that may have gone unmentioned in the text.

By applying our proposed graph extraction pipeline to articles grouped by timestamp windows across a chronological sequence, we can curate a temporal graph dataset, which is a dataset that enables the use of temporal graph representation learning methods. Such a dataset can be used to predict future business networking interactions through inductive link prediction over graphs.

Moreover, another compelling utilization of the graph resulting from our proposed method is the extraction of trading-related content from news texts originating from diverse international markets. For instance, by conducting pairwise graph isomorphism tests on subgraphs composed of organization nodes and their respective neighborhoods from either of the generated graphs, we could effectively identify similar companies operating in different markets.

These extended applications of our methodology underscore its versatility and potential impact in facilitating a deeper understanding of dynamic business networks and inter-market relationships.

5.2. Current Limitations

Our methodology can generate graphs with sufficient quality to be used for business networking analysis at low computation costs. Yet two noticeable limitations can be identified within the current iteration of our methodology.

Extracting relations with OpenIE is primarily a rule-based and heuristic-driven approach. Its performance depends on the quality of linguistic patterns and heuristics used for relation extraction. While it can extract valuable structured information from text, it may not capture complex relationships described with more than one sentence or other nuances in the same way that more advanced large language models are capable of.

A relation classification accuracy score of 84% shows the effective ability of grouping relationship types before applying clustering models to classify relationships based on word vector representation without annotations. Yet this score is well below the state-of-the-art score in stand-alone relation classification tasks that reach 97% on partially annotated datasets such as Few-Rel while relying on large language models and few-shot learning [46].

The integration of large language models in the extraction of a business network knowledge graph based on our ontology is the primary concern of our next iteration. This will come at the cost of high computation and other drawbacks described earlier in Section 3.3.

6. Conclusions

Effectively transforming business news into structured networks for analysis presents significant opportunities for business networking and thus motivated this work. This study introduces TraCER, a pipeline that solves the task of converting unstructured news text into a graph. The pipeline, composed of key subtasks including text preprocessing, named entity recognition, entity-linking, triple extraction, and relation classification, demonstrated the ability to automatically build a domain-specific knowledge graph for the analysis of business networking from the Reuters corpus. TraCER is computationally efficient and based on interpretable models and heuristics.

TraCER opens possibilities including business-to-business interaction predictions and business type predictions. Beyond these applications, TraCER can enable tracking chronological business interactions and matching businesses across different markets. However, the current stage of our study presents some limitations. One limitation is the inability to identify relationships described with multiple sentences. This paves the road for future works that will involve the use of large language models in the construction of business networking knowledge graphs.

Author Contributions: Conceptualization, D.G. and K.K.; methodology, D.G.; software, D.G.; validation, D.G. and K.K.; formal analysis, D.G.; investigation, D.G.; resources, K.K.; data curation, D.G.; writing—original draft preparation, D.G.; writing—review and editing, D.G. and K.K.; visualization, D.G.; supervision, K.K.; project administration, D.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding

Data Availability Statement: All the experimental source code is available at the following repositories: (https://github.com/semlab/tracer, accessed on 27 December 2023), (https://github.com/semlab/triplex, accessed on 27 December 2023). The Reuters-21578 dataset is freely accessible throughout the web (https://kdd.ics.uci.edu/databases/reuters21578/reuters21578.html, accessed on 27 December 2023).

Acknowledgments: We would like to acknowledge the annotation work from the students in our department which allowed us to test the accuracy of our heuristics.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

OpenIE	Open information extraction
NLP	Natural language processing

Appendix A

In this appendix, we offer a focused examination of a labeled graph visual extracted from a corresponding sampled article of the Reuters dataset, providing a "zoomed-in" perspective. This highlights the alignment between the textual content and the automatically extracted graphical representations.

For this detailed demonstration, the sampled article from the Reuters-21578 corpus is titled "WALL STREET STOCKS/TENNECO INC" and was published on 10 March 1987. Below is an excerpt of the article after preprocessing.

> Tenneco Inc., a company that has long been rumored to be a takeover candidate, rose sharply today when speculation surfaced that investor T. Boone Pickens may be targeting the company for an acquisition, traders and analysts said. Tenneco spokesman Joseph Macrum said "we have no comment to make whatsoever." Pickens was not available for comment. Traders noted that activity in the stock increased today after a published report linked Pickens to Tenneco. Tenneco rose two points to 48-3/4. Paul Feretti, an analyst with New Orleans-based Howard, Weil, Labouisse, Friedrichs, Inc, said he was not surprised at market rumors that Tenneco might be the target of a takeover attempt. "It's pure market speculation that Boone Pickens and their group may be interested," Feretti said. "However, Tenneco would be a challenge to run because of its sheer size and diversity. Mr. Pickens is a man who likes a challenge[...]"

From the text, the following extracted triples to which the coreference resolution, as well as the filtering and the linking heuristics, have been applied are presented in Table A1.

From the triples, the named entities are isolated and used to classify the relationships into a category from the ontology by employing the clustering mechanism proposed. Table A2 shows named entity pairs involved in a triple relationship and their corresponding assigned type.

Table A1. Triples extracted from the sample article.

Subject	**Relation**	**Object**
Joseph Macrum	be spokesman of	Tenneco
Tenneco	hold	gas reserve
Feretti	estimate	Tenneco's breakup value
Feretti	conservatively estimate	Tenneco's breakup value

Table A2. Type and named entity involved in extracted relationships and their respective assigned relationship type.

Source Entity	**Destination Entity**	**Relationship Category**
PERSON:Joseph Macrum	ORGANIZATION:Tenneco	collaborates
ORGANIZATION:Tenneco	PRODUCT:gas	produces
PERSON:Feretti	ORGANIZATION:Tenneco	collaborates
PERSON:Feretti	ORGANIZATION:Tenneco	collaborates

We can observe that even though some names are mentioned in the text, they are not part of the extracted named entities. This happens because they are likely not appearing in OpenIE extractions that respect our ontology rules; thus, they end up discarded. This might be a limitation as discussed in Section 5.2 if a relationship is described with more than one sentence. This is planned to be addressed in future work with the use of language models.

From the identified entities and categorized relationships, we can represent the graph. Figure A1 presents the resulting graph for the sampled article.

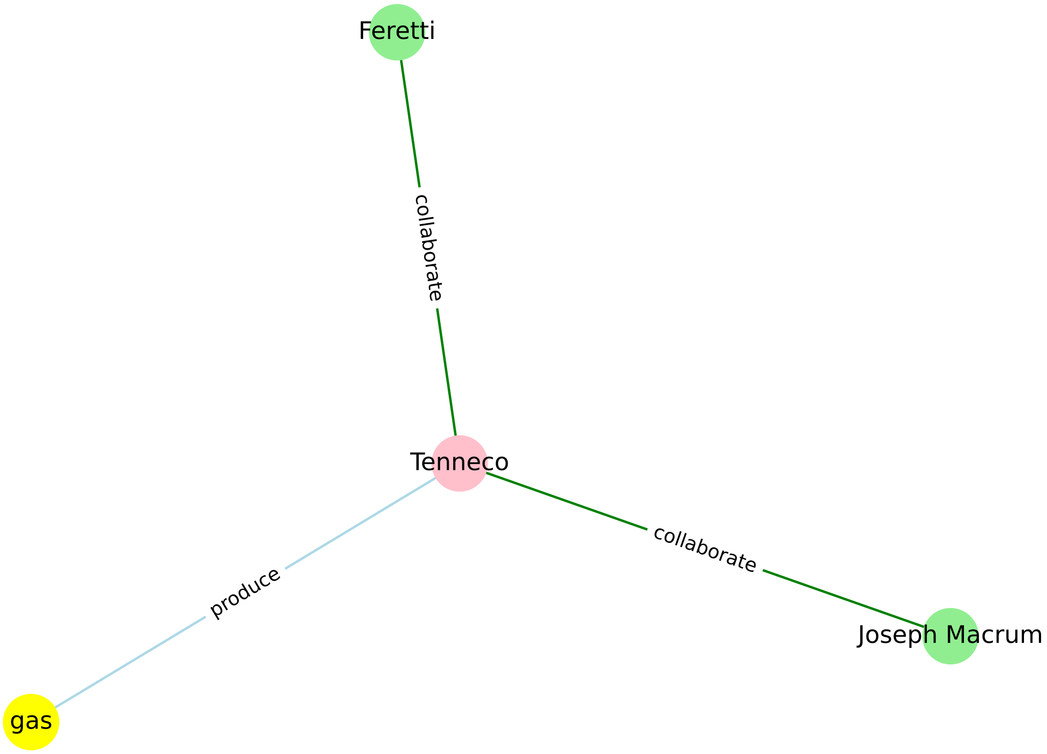

Figure A1. Visualization of the graph extracted from the sample article.

This detailed perspective on the results further underscores the efficacy of our automated graph extraction approach in capturing and visualizing the key insights within the textual data for business networking and emphasizing the clarity of our research findings.

Appendix B

Here, we provide a couple of examples for the linking heuristic described in Section 3.5. The examples consist of sentences, a candidate triple extraction with associated named entity types, and whether or not they represent an instance of entity link.

Example 1

> Sales of previously owned homes dropped 14.5 pct in January to a seasonally adjusted annual rate of 3.47 mln units, the National Association of Realtors (NAR) said.

The candidate triple (National Association of Realtors [ORGANIZATION], be, NAR [ORGANIZATION]) matches the definition; thus, NAR is identified as being the same entity as the National Association of Realtors.

Example 2

> Kevlar was invented by Du Pont in the late 1960s and is five times stronger than steel and 10 times stronger than aluminum on an equal wieght basis, and is used to replace metals in a variety of products, according to the company.

The candidate triple (Kevlar [PERSON], be, Du Pont [ORGANIZATION]) is an example that does not match the definition; thus, this triple will be later filtered and classified as an edge in the graph.

Appendix C

We conduct an ablation study to understand the impact of key steps in our proposed pipeline. The ablation consist of comparing the number of extracted entities and relationships after using different configurations of our pipeline on the Reuters corpus. The configurations include the full TraCER pipeline, the TraCER pipeline without the implementation of co-reference resolution (no coref), and the TraCER pipeline without the entity linking heuristic (no link). Table A3 summarizes the extractions results from which we can observe the contribution of each of those steps in reducing noises and redundancies in the resulting graph.

Table A3. Number of extracted entities and relationships for different configurations of the TraCER pipeline.

Configurations	Entities	Relationships
TraCER (complete)	6761	9161
TraCER (no coref)	6756	9831
TraCER (no link)	8082	13,842

Another configuration was to remove the relation categorization from the full pipeline to inspect the number of unique relationships. Without categorizing the relationship, the pipeline produces 3379 unique relationships, from which a large portion of different instances appear fewer than ten times. This would have been an handicap for link prediction algorithms without the addition of the categorization mechanism that reduces the number of unique links to five.

References

1. Xia, F.; Sun, K.; Yu, S.; Aziz, A.; Wan, L.; Pan, S.; Liu, H. Graph learning: A survey. *IEEE Trans. Artif. Intell.* **2021**, *2*, 109–127. [CrossRef]
2. Hogan, A.; Blomqvist, E.; Cochez, M.; d'Amato, C.; Melo, G.d.; Gutierrez, C.; Kirrane, S.; Gayo, J.E.L.; Navigli, R.; Neumaier, S.; et al. Knowledge graphs. *ACM Comput. Surv. CSUR* **2021**, *54*, 1–37.
3. Bronstein, M.M.; Bruna, J.; Cohen, T.; Veličković, P. Geometric deep learning: Grids, groups, graphs, geodesics, and gauges. *arXiv* **2021**, arXiv:2104.13478.
4. Jumper, J.; Evans, R.; Pritzel, A.; Green, T.; Figurnov, M.; Ronneberger, O.; Tunyasuvunakool, K.; Bates, R.; Žídek, A.; Potapenko, A.; et al. Highly accurate protein structure prediction with AlphaFold. *Nature* **2021**, *596*, 583–589. [CrossRef] [PubMed]
5. Chunaev, P. Community detection in node-attributed social networks: A survey. *Comput. Sci. Rev.* **2020**, *37*, 100286. [CrossRef]
6. Li, Z.; Chen, X.; Song, J.; Gao, J. Adaptive label propagation for group anomaly detection in large-scale networks. *IEEE Trans. Knowl. Data Eng.* **2022**, *35* , 12053–12067. [CrossRef]
7. Guo, Q.; Zhuang, F.; Qin, C.; Zhu, H.; Xie, X.; Xiong, H.; He, Q. A survey on knowledge graph-based recommender systems. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 3549–3568. [CrossRef]
8. Chen, X.; Jia, S.; Xiang, Y. A review: Knowledge reasoning over knowledge graph. *Expert Syst. Appl.* **2020**, *141*, 112948. [CrossRef]
9. Hu, W.; Fey, M.; Zitnik, M.; Dong, Y.; Ren, H.; Liu, B.; Catasta, M.; Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 22118–22133.
10. Li, J.; Sun, A.; Han, J.; Li, C. A Survey on Deep Learning for Named Entity Recognition. *IEEE Trans. Knowl. Data Eng.* **2022**, *34*, 50–70. [CrossRef]
11. Nasar, Z.; Jaffry, S.W.; Malik, M.K. Named Entity Recognition and Relation Extraction: State-of-the-Art. *ACM Comput. Surv.* **2021**, *54*, 1–39. [CrossRef]
12. Zachary, W. An Information Flow Model for Conflict and Fission in Small Groups. *J. Anthropol. Res.* **1976**, *33*, 452–473. [CrossRef]
13. Perozzi, B.; Al-Rfou, R.; Skiena, S. DeepWalk: Online Learning of Social Representations. In Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA, 24–27 August 2014; pp. 701–710. [CrossRef]
14. Kipf, T.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the 5th International Conference on Learning Representations, ICLR 2017—Conference Track Proceedings, Toulon, France, 24–26 April 2017 ; pp. 1–14.
15. Miller, G.A. WordNet: A Lexical Database for English. *Commun. ACM* **1995**, *38*, 39–41. [CrossRef]
16. Lenat, D.B. CYC: A Large-Scale Investment in Knowledge Infrastructure. *Commun. ACM* **1995**, *38*, 33–38. [CrossRef]
17. Giles, C.; Bollacker, K.; Lawrence, S. CiteSeer: An automatic citation indexing system. In Proceedings of the ACM International Conference on Digital Libraries, ACM, Pittsburgh, PA, USA, 23–26 June 1998; pp. 89–98.
18. Zhong, L.; Wu, J.; Li, Q.; Peng, H.; Wu, X. A Comprehensive Survey on Automatic Knowledge Graph Construction. *arXiv* **2023**, arXiv:2302.05019.
19. Kertkeidkachorn, N.; Ichise, R. An Automatic Knowledge Graph Creation Framework from Natural Language Text. *IEICE Trans. Inf. Syst.* **2018**, *E101.D*, 90–98. [CrossRef]
20. Sant'Anna, D.T.; Caus, R.O.; dos Santos Ramos, L.; Hochgreb, V.; dos Reis, J.C. Generating Knowledge Graphs from Unstructured Texts: Experiences in the E-commerce Field for Question Answering. In Proceedings of the Joint Proceedings of Workshops AI4LEGAL2020, NLIWOD, PROFILES 2020, QuWeDa 2020 and SEMIFORM2020, Colocated with the 19th International Semantic Web Conference (ISWC 2020), CEUR, Virtual Conference, 1–6 November 2020; pp. 56–71.
21. Yu, S.; He, T.; Glass, J. AutoKG: Constructing Virtual Knowledge Graphs from Unstructured Documents for Question Answering. *arXiv* **2021**, arXiv:2008.08995
22. Saha, S.; Mausam. Open Information Extraction from Conjunctive Sentences. In Proceedings of the 27th International Conference on Computational Linguistics, Santa Fe, NM, USA, 21–24 August 2018; pp. 2288–2299.
23. Devlin, J.; Chang, M.W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Minneapolis, MN, USA, 2–7 June 2019 2019; pp. 4171–4186. [CrossRef]
24. Zhang, Z.; Han, X.; Liu, Z.; Jiang, X.; Sun, M.; Liu, Q. ERNIE: Enhanced Language Representation with Informative Entities. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, Florence, Italy, 28 July–2 August 2019; pp. 1441–1451. [CrossRef]
25. Cao, E.; Wang, D.; Huang, J.; Hu, W. Open Knowledge Enrichment for Long-Tail Entities. In Proceedings of the Web Conference 2020, Taipei, Taiwan, 20–24 April 2020; Association for Computing Machinery: New York, NY, USA; Springer: Cham, Switzerland, April 2020; pp. 384–394.
26. Ruan, T.; Xue, L.; Wang, H.; Hu, F.; Zhao, L.; Ding, J. Building and Exploring an Enterprise Knowledge Graph for Investment Analysis. In *Proceedings of the Semantic Web—ISWC 2016, Kobe, Japan, 17–21 October 2016*; Groth, P., Simperl, E., Gray, A., Sabou, M., Krötzsch, M., Lecue, F., Flöck, F., Gil, Y., Eds.; Springer: Cham, Switzerland, 2016; pp. 418–436.
27. Dai, L.; Yin, Y.; Qin, C.; Xu, T.; He, X.; Chen, E.; Xiong, H. Enterprise Cooperation and Competition Analysis with a Sign-Oriented Preference Network. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, 6–10 July 2020; pp. 774–782.

28. Hillebrand, L.; Deußer, T.; Dilmaghani, T.; Kliem, B.; Loitz, R.; Bauckhage, C.; Sifa, R. Kpi-bert: A joint named entity recognition and relation extraction model for financial reports. In Proceedings of the 2022 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, 21–25 August 2022; pp. 606–612.
29. Gohourou, D.; Kuwabara, K. Building a Domain-Specific Knowledge Graph for Business Networking Analysis. In *Proceedings of the Intelligent Information and Database Systems: 13th Asian Conference, ACIIDS 2021, Phuket, Thailand, 7–10 April 2021*; Proceedings 13; Springer: Cham, Switzerland, 2021; pp. 362–372.
30. Grüninger, M.; Fox, M. Methodology for the Design and Evaluation of Ontologies. In Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, Montreal, QC, Canada, 13 April 1995.
31. Noy, N.F.; McGuinness, D.L. *Ontology Development 101: A Guide to Creating Your First Ontology*; Stanford University: Stanford, CA, USA, 2001.
32. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. 2018 *Preprint*. Available online: https://paperswithcode.com/paper/improving-language-understanding-by (accessed on 27 December 2023).
33. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I. Language models are unsupervised multitask learners. *OpenAI Blog* **2019**, *1*, 9.
34. Brown, T.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.D.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 1877–1901.
35. Chowdhery, A.; Narang, S.; Devlin, J.; Bosma, M.; Mishra, G.; Roberts, A.; Barham, P.; Chung, H.W.; Sutton, C.; Gehrmann, S.; et al. Palm: Scaling language modeling with pathways. *arXiv* **2022**, arXiv:2204.02311.
36. Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. Llama: Open and efficient foundation language models. *arXiv* **2023**, arXiv:2302.13971.
37. Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G.S.; Dean, J. Distributed Representations of Words and Phrases and their Compositionality. *Adv. Neural Inf. Process. Syst.* **2013**, *26* , 3111–3119.
38. Mikolov, T.; Chen, K.; Corrado, G.S.; Dean, J. Efficient Estimation of Word Representations in Vector Space. *arXiv* **2013**, arXiv:1301.3781.
39. Angeli, G.; Premkumar, M.J.; Manning, C.D. Leveraging linguistic structure for open domain information extraction. In *ACL-IJCNLP 2015—53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, Proceedings of the Conference*; Association for Computational Linguistics: Beijing, China, 2015; Volume 1, pp. 344–354. [CrossRef]
40. Klein, D.; Smarr, J.; Nguyen, H.; Manning, C.D. Named entity recognition with character-level models. In Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, Edmonton, AB, Canada, 31 May 2003; pp. 180–183.
41. Mikolov, T.; Yih, W.t.; Zweig, G. Linguistic Regularities in Continuous Space Word Representations. In Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Atlanta, GA, USA, 31 May–1 June 2013; pp. 746–751.
42. Řehůřek, R.; Sojka, P. Software Framework for Topic Modelling with Large Corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks, Valletta, Malta, 25 July 2010; pp. 45–50. Available online: http://is.muni.cz/publication/884893/en (accessed on 20 November 2023)
43. Manning, C.; Surdeanu, M.; Bauer, J.; Finkel, J.; Bethard, S.; McClosky, D. The Stanford CoreNLP Natural Language Processing Toolkit. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Baltimore, MD, USA, 23–24 June 2014; pp. 55–60. [CrossRef]
44. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
45. Wu, L.; Petroni, F.; Josifoski, M.; Riedel, S.; Zettlemoyer, L. Scalable Zero-shot Entity Linking with Dense Entity Retrieval. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Online, 16–20 November 2020; pp. 6397–6407. [CrossRef]
46. Soares, L.B.; FitzGerald, N.; Ling, J.; Kwiatkowski, T. Matching the blanks: Distributional similarity for relation learning. *arXiv* **2019**, arXiv:1906.03158.