UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Universitat Politècnica de Catalunya
Teoria del Senyal i Comunicacions

This thesis is submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy (PhD)

Knowledge Graph Population from News Streams

---

by Dèlia Fernàndez Cañellas

Advisor: Elisenda Bou-Balust
Advisor: Xavier Giró-i-Nieto
Tutor: Ferran Marques
Barcelona, July 2023

Abstract

Media producers publish large amounts of multimedia content online - both text, audio, image and video. As the online media market grows, the management and delivery of contents becomes a challenge. Semantic and linking technologies can be used to organize and exploit these contents through the use of knowledge graphs. This industrial doctorate dissertation addresses the problem of constructing knowledge resources and integrating them into a system used by media producers to manage and explore their contents. For that purpose, knowledge graphs and their maintenance through Information Extraction (IE) from news streams is studied. This thesis presents solutions for multimedia understanding and knowledge extraction from online news, and their exploitation in real product applications, and it is structured in three parts.

The first part consists on the construction of IE tools that will be used for knowledge graph population. For that, we built an holistic Entity Linking (EL) system capable of combining multimodal data inputs to extract a set of semantic entities that describe news content. The EL system is followed by a Relation Extraction (RE) model that predicts relations between pairs of entities with a novel method based on entity-type knowledge. The final system is capable of extracting triples describing the contents of a news article.

The second part focuses on the automatic construction of a news event knowledge graph. We present an online multilingual system for event detection and comprehension from media feeds, called VLX-Stories. The system retrieves information from news sites, aggregates them into events (event detection), and summarizes them by extracting semantic labels of its most relevant entities (event representation) in order to answer four Ws from journalism: who, what, when and where. This part of the thesis deals with the problems of Topic Detection and Tracking (TDT), topic modeling and event representation.

The third part of the thesis builds on top of the models developed in the two previous parts to populate a knowledge graph from aggregated news. The system is completed with an emerging entity detection module, which detects mentions of novel people appearing on the news and creates new knowledge graph entities from them. Finally, data validation and triple classification tools are added to increase the quality of the knowledge graph population.

This dissertation addresses many general knowledge graph and information extraction problems, like knowledge dynamicity, self-learning, and quality assessment. Moreover, as an industrial work, we provide solutions that were deployed in production and verify our methods with real customers.

Keywords: Knowledge Graph Population, Natural Language Processing, Information Extraction, Entity Linking, Named Entity Recognition, Named Entity Disambiguation, Relation Extraction, Topic Detection and Tracking, Topic Modeling, Triple Validation

Acknowledgments

This thesis is the result of a long journey that has made me grow both as researcher and a person, and I would like to thank everyone who has been part of it.

First of all, I want to thank my two advisors, Elisenda Bou and Xavier Giró, without whom this thesis would have not been possible. I want to thank Elisenda Bou for the big opportunity of working in a company like Vilynx, and doing my PhD in it. Her bright ideas have guide my work into reaching goals I would have never imagined. I am also extremely thankful to Xavier Giró for his support and passion guiding students, including myself, and encouraging them to pursue their dreams. Without his enthusiasm, hard work and help this thesis would have never seen the light. I would also like to thank my tutor Ferran Marques for all his support, help and comprehension thought the years that helped me pursue my PhD.

I would like to acknowledge all the colleagues at Vilynx, who have made this journey more fun and easier to bear. Specially I would like to thank Joan Espadaler, as my coworker and friend, with whom I shared a big part of this journey. I also want to acknowledge and thank Juan Carlos Riveiro, as Vilynx's CEO, who supported my persuasion for a PhD and encouraged our ambitious goals. I do not want to forget all the team involved in this thesis contributions. First, I want to thank my closest coworkers and contributors: Blai Garolera, Aleix Colom, Joan Marco Rimmek, David Rodriguez, Gemma Canet, Adrià Barja, Isabel Fernandez and Albert Renau, who supported me both in the past and present. Also, I want to give special thanks to Marc Codina and Marc Sastre for their work on the knowledge graph; David Varas, Issey Masuda and Alejandro Woodward for their work on image recognition and speech to text systems; Jordi Ferreira for his dashboard and UI work that made our work shine; and Asier Aduriz for his advice on scalability issues.

Thanks also to all the friends in Barcelona with whom I know I can always count and have supported me on the difficult times.

Finally, and most important, I can not finish without acknowledging and thanking the unconditional support from my parents, Artur and Mariantonia, whose love and guidance are with me in whatever I pursue. I want to thank them for teaching me to follow my dreams, bringing me the opportunities to make them happen. Thanks also for giving me all the emotional support I needed in the good and bad times and encouraging my decisions.

Contents

List of Figures

List of Tables

Introduction

Today's media and news organizations are constantly generating large amount of multimedia content which is majorly delivered online. Computational approaches can govern content creation, production, search, and the promotion and distribution of this content to different audiences by automatically indexing and linking it to other contents, trends on social networks or events reported in the news. With the increasing adoption of Machine Learning (ML) and Deep Learning (DL) in Natural Language Processing (NLP), and the standardization of semantic technologies it is a particularly exciting time to investigate and generate new solutions.

Content generated by media companies, whether it is in natural language text, images, video or audio, is unstructured. Nevertheless, computers require data to be expressed in a machine-readable (structured) format in order to exploit the knowledge embedded on it and build intelligent applications, like search and recommendations. Manually extracting this structured knowledge from contents is extremely expensive and unfeasible in practice, therefore automatic methods for large-scale content understanding are needed. Moreover, contents in the Web are presented in a multilingual fashion. It is needed, thus, some language agnostic representation in order to link such multilingual contents.

Semantic technologies offer interesting solutions to establish universal representations of real world entities and its relations through the use of **knowledge graphs** (KGs). Knowledge graphs store collections of facts about people, things, and places in the world and the relations between them. During the last decade, different research communities have put great effort into building several noteworthy large knowledge graphs, like Freebase [25], Wikidata [197], DBPedia [15], OpenCyc [116], YAGO [188], *Google Knowledge Graph*[1], Facebook's *Social Graph*[193] and *IBM Watson's* [64]. These systems are already widely used and important for many information retrieval tasks, and have been robustly studied by the Semantic Web community.

With much of human knowledge residing in books and other text documents, knowledge graphs have usually been constructed from processing natural language text. **Information Extraction** (IE) is the NLP sub-field in charge of encoding text in a machine-readable representation, i.e. transforming unstructured natural language text to strutctured knowledge. IE techniques like Entity Linking (EL) and Relation Extraction (RE) can be used for extracting useful metadata describing text contents. Nevertheless, content generated by media companies is often multimodal. Understanding entities and relations in a multimodal manner allows for a more complete representation of the world and thus a more robust system: multimodal data produces complementary and correlated information, which provides additional redundancy for better robustness. There is still a small number of knowledge graphs exploiting the visual part, but some are already

[1] https://blog.google/products/search/introducing-knowledge-graph-things-not/

showing improvements and promising results for visual question answering [226], scene classification and object detection[33], and visual reasoning [225].

One of the main problems with knowledge graphs is the *knowledge acquisition bottleneck*, i.e. they are often constructed manually, using experts with specific domain knowledge for the field of interest. However, real world is dynamic and it constantly grows on relevant entities and changing relations. Specifically, in the field of news, events reported often involve changes in relations and unknown entities that are not captured by these resources, and are therefore missed by most knowledge graphs. Detecting these out-of-knowledge-graph (OOKG) facts and their related emerging entities is crucial for any knowledge graph maintenance process [87, 127, 168]. In particular, when willing to provide efficient tools for media applications. Knowledge graph population is the task of automatically doing this process, which usually encompasses IE techniques to extract novel entities and facts.

In this thesis we address the creation and adoption of knowledge graphs to be used in media applications. To do that, we deal with the aforementioned problems by proposing a dynamic knowledge graph system which learns and updates entities and relations based on the information extracted from aggregated news streams. News stories provide information about real-world events and its progress in real time, containing thus emerging entities and changes in relations. Moreover, working with information extracted from already aggregated news provides redundancy between data, which helps into making more robust decisions and thus building a more robust system. This is an industrial PhD thesis, developed in an industrial context in Vilynx[2] company. All the work presented is deployed in production services and used by Vilynx's.

To create this system capable of providing advanced knowledge services which understands and learns from the real world, we divide the work in the three parts in which this thesis is structured: a) construction of IE tools to extract entities and relations from text, and integrate multi-modal information for a more robust disambiguation, b) detection of aggregated news and creation of an event-based knowledge graph, and finally c) population of a knowledge graph with emerging entities and new facts detected from the aggregated news, and its validation to generate high-quality data.

0.1 Objectives

In this section we define which are the objectives of this dissertation. The main goal of the thesis is to **create a system capable of providing advanced knowledge services which understand and learn from the real world**. To do that, we approach the creation and population of a knowledge graph from aggregated news articles. This system will encompass the construction of IE tools together with a real-time news event detection and representation framework.

The complete system presented in this dissertation is schematized in Figure 0.1, where the three sections of this thesis can be discerned. Now we enumerate the main objective for each one of them:

- **Part I:** The goal of this part is the construction of IE tools capable of extracting

[2] https://www.vilynx.com

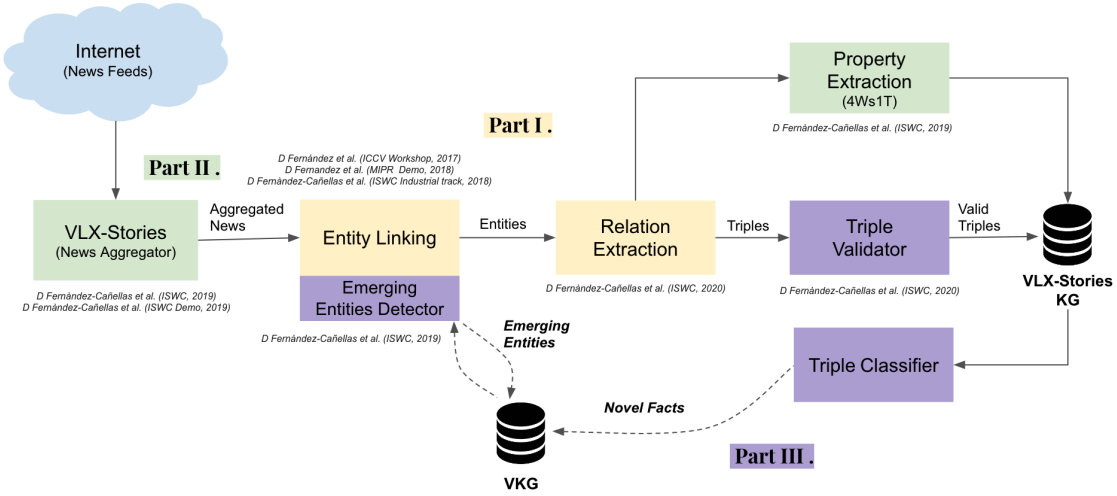


Figure 0.1: Scheme of the complete system presented in this dissertation. Part I consists on the construction of IE tools for entity linking and relation extraction (yellow modules). Part II is the creation of a real-time news aggregator, called VLX-Stories, which performs clustering over articles parsed from news feeds collected from the internet. IE tools presented in Part I will be used to extract metadata from the aggregated articles and a property extraction module will be used to understand the 4Ws ("*who*", "*what*", "*where*" and "*when*") of the news story, plus the "*topic*", to populate an event knowledge graph, called VLX-Stories KG (green modules). Finally, Part III consists on the discovery of emerging entities and the validation of novel facts from the aggregated news stories, to populate VKG (purple modules).

entities and relations from text. We can divide our contributions into two parts:

  - We aim to create an Entity Extraction system which combines multi-modal information for a more robust Entity Linking.
  - We aim to create a Relation Extraction system which encodes semantic information to enhance its performance.

- **Part II**: The goal of this part is two-fold:

  - First, we aim to detect events through the construction of a news aggregator system. This module will deal with the tasks of topic modeling and Topic Detection and Tracking (TDT).
  - Second, we aim to construct an Event knowledge graph, called VLX-Stories KG. We will study the construction of an Event Ontology, inspired on the four journalistic questions of the "who", "what", "where" and "when", and the extraction of this information.

- **Part III**: The goal of this last part is the population of the general knowledge graph presented in Part I, with the emerging entities and novel facts detected from the aggregated news, and the validation of these facts to provide high quality triple population.

0.2 Research Question

With the growing amount of unstructured contents being generated daily by media companies, the need for indexing such contents in an structured representation arises. Knowledge graph's entities can be used for tagging such contents, but most knowledge graphs are highly incomplete, limiting their applicability. On the other hand, many of those missing entities and facts are mentioned in the large amount of unstructured text available through news published in the Internet. In this context, the research conducted in this dissertation can be understood as approaching an answer to the following question: **Can we generate a dynamic knowledge graph that updates according to world changes by leveraging on the information in the news?** Therefore, this thesis tackles the challenge of automatically extracting information from news to populate a knowledge graph. As the scope of such research question is broad, we narrow our focus into building three interconnected modules, with its own research questions.

Part I of this thesis focuses on the extraction of information from media contents, and is divided into the two main sub-tasks of IE: entity extraction and relation extraction. First, an entity extraction system is created and we tackle two main questions: **Can we use multi-modal information to enhance entity extraction?**, and **Can textual information (description, title) be used to label video contents?**. Once entities are extracted, relations between such entities are predicted in a relation extraction system. To develop such system we investigate on: **Can we use semantic information to improve relation extraction systems?**.

Part II pursues the detection of relevant news events and the extraction of information from it. In this part the next questions are tackled: **Can we create systems that automatically detect events?** and **How can we extract and store semantic representations of news events?**. This part will use the knowledge graph and IE systems developed on the previous two parts to generate semantic representation of the events.

Finally, Part III combines the three previous parts for the detection of novel entities and facts. We address the following question: **Can we automatically detect world changes?**. Previous models will be combined with filters and validation tools in order to populate the knowledge graph with relevant and reliable data.

0.3 Major Contributions

All the technical contributions presented in this dissertation have been published at peer-reviewed venues or patented in the U.S.. Moreover, two Master thesis have been advised.

Peer-reviewed Publications

- Fernàndez-Cañellas, Dèlia, Joan Marco Rimmek, Joan Espadaler, Blai Garolera, Adrià Barja, Marc Codina, Marc Sastre, Xavier Giro-i-Nieto, Juan Carlos Riveiro, and Elisenda Bou-Balust. "**Enhancing Online Knowledge Graph Population with Semantic Knowledge**." In International Semantic Web Conference, pp. 183-200. Springer, Cham, 2020.

- Fernàndez-Cañellas, Dèlia, Joan Espadaler, David Rodriguez, Blai Garolera,

Gemma Canet, Aleix Colom, Joan Marco Rimmek, Xavier Giro-i-Nieto, Elisenda Bou, and Juan Carlos Riveiro. "**VLX-stories: Building an online event knowledge base with emerging entity detection**." In International Semantic Web Conference, pp. 382-399. Springer, Cham, 2019.

- Fernández Cañellas, Dèlia, Joan Espadaler, Blai Garolera, David Rodriguez, Gemma Canet, Aleix Colom, Joan Marco Rimmek, Xavier Giró Nieto, Elisenda Bou Balust, and Juan Carlos Riveiro. "**VLX-Stories: a semantically linked event platform for media publishers**." In Proceedings of the ISWC 2019 Satellite Tracks (Posters Demonstrations, Industry, and Outrageous Ideas): co-located with 18th International Semantic Web Conference (ISWC 2019): Auckland, New Zealand, October 26-30, 2019, pp. 233-236. CEUR-WS. org, 2019.

- Fernàndez, Dèlia, Elisenda Bou Balust, Xavier Giró Nieto, Juan Carlos Riviero, Joan Espadaler, David Rodriguez, Aleix Colom Serra, Joan Marco Rimmek, David Varas, Issey Massuda, Carlos Roig. "**Linking Media: adopting Semantic Technologies for multimodal media connection**." Proceedings of the ISWC 2018 Posters Demonstrations, Industry and Blue Sky Ideas Tracks co-located with 17th International Semantic Web Conference (ISWC 2018): Monterey, USA: October 8th to 12th, 2018, pp. 1-2. CEUR-WS. org, 2018.

- Fernandez, Delia, Joan Espadaler, David Varas, Issey Masuda, Aleix Colom, David Rodriguez, David Vegas, Miquel Montalvo, Xavi Giro-i-Nieto, Juan Carlos Riveiro, Elisenda Bou. "**What is going on in the world? A display platform for media understanding**." In 2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), pp. 204-205. IEEE, 2018.

- Fernández, Delia, David Varas, Joan Espadaler, Issey Masuda, Jordi Ferreira, Alejandro Woodward, David Rodríguez, Xavier Giró-i-Nieto, Juan Carlos Riveiro, and Elisenda Bou. "**Vits: video tagging system from massive web multimedia collections**." In Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 337-346. 2017.

Patents Granted

- Elisenda Bou Balust, Juan Carlos Riveiro Insua, Delia Fernandez Cañellas, Joan Espadaler Rodes, Asier Aduriz Berasategi, and David Varas Gonzalez. "**Video tagging system and method**." U.S. Patent 11,256,741, issued February 22, 2022.

Patent applications [3]

- Juan Carlos Riveiro, Elisenda Bou, Delia Fernandez, Joan Espadaler et al. "**Event Knowledge Base with Emerging Entity Detection.**" U.S. Patent Application No. 62/853,047, issued May 26, 2019.

- Juan Carlos Riveiro, Elisenda Bou, Delia Fernandez, Joan Espadaler et al. "**Entity Based Content Processing**." U.S. Patent Application No. 16/566,635, issued October 9, 2019.

[3] Following patents were applied by Vilynx SL. but were not completed due to Apple Inc. acquisition.

- Juan Carlos Riveiro, Elisenda Bou, Delia Fernandez, Joan Espadaler et al. "**Self-learning Knowledge Graph**, 2020." [4]

MSc Co-directed

- Joan Marco-Rimmek, "**BERT-based Neural Relation Extraction with Distant Supervision**", 2020. Co-directed by Bou Balust, Elisenda; Fernández, Dèlia; Peguroles Balles, Josep Rafael.
- Adrià Barja Romero, "**Knowledge Graph Representations for Entity Disambiguation**", 2020. Co-directed by Bou Balust, Elisenda; Fernández, Dèlia; Peguroles Balles, Josep Rafael.

[4] This patent application was never submitted do to Apple Inc. acquisition interfered.

Knowledge Graph

Since its inception by Google on 2012, the term knowledge graph has emerged as a major interest area in Artificial Intelligence (AI) for both research and industry. Google presented the *Google Knowledge Graph* with the motto of "making search about things not strings". Since then, other companies have invested in knowledge graphs, and several KG-centric startups have emerged in multiple countries and continents. Nowadays many well known AI-based products and applications are powered by knowledge graphs, like personal assistants (e.g. Alexa, Siri and Google Assistant) or eCommerce tools (e.g. eBay and Amazon).

In this part we will review some basic theory on knowledge graphs (0.4), semantic technologies (0.5), public knowledge graphs (0.6), and describe the media based knowledge graph used in this dissertation, called *Vilynx Knowledge Graph* (VKG) (0.7). VKG will be the core part of the technologies developed.

0.4 Background on Knowledge Graphs

As described by Kejriwal [104], *a knowledge graph is a graph-theoretic representation of human knowledge such that it can be ingested with semantics by a machine.* In other words, it is a way to express 'knowledge' using graphs, in a for which a machine would be able to conduct reasoning and inference to answer questions in some meaningful way.

A knowledge graph typically contains an ontology (0.4.1), defining the semantic model, and the data itself (0.4.2). In this Section we will describe the two parts of a knowledge graph and its related components.

0.4.1 Ontology

Ontologies have become increasingly important with the use of knowledge graphs. The term ontology was defined by Gruber in 1993 as a *specification of a conceptualization* [74]. Later, on 2001, Noy and McGuiness presented a more operational description of an ontology being a *formal, explicit description of concepts in a domain of discourse, properties of each concept describing various features and attributes of the concepts, and restrictions on slots* [142]. The ontology provides, thus, the schema and rules for interpretation of the entities and facts comprising the domain knowledge. Even ontologies may have some structural differences. These are the main shared components needed to define an ontology:

- **Types (classes or categories):** are abstract objects that are defined by values of aspects that are shared by members of a class [13]. For example classes may

differentiate individuals with categories or types like "person", "organization", "location", etc. These classes can be related hierarchically being for example a root type "location" a container of more specific types like "country", "city" and "building". Individuals in a category are known to have a set of properties or relations related to it, as well as a set of constraints.

- **Entities (instances)**: are basic objects representations within a domain [158]. Grammatically, entities tend to be nouns or objects mentioned in sentences and represented by classes like "person", "location" or "organization". For example, the English rock band "*The Beatles*", or the 44th president of the United States, "*Barack Obama*", would be entities from a knowledge graph.

- **Relations (properties or predicates)**: relations define how objects in an ontology relate to each other. Typically a relation is of a particular type (or class) that specifies in what sense the object is related to the other object in the ontology. Thus, relations can have a hierarchy being a relation "cast member" a super-class of the relations "actor" and "director".

- **Axioms (rules or constraints)**: are formal descriptions of what must be true in order for some assertion to be accepted as input. For example, we can define that an instance of type "person" can only have a property "birth date", or that two entities related as "siblings" can not be related as "father" and "child".

0.4.2 Data

The information queried from a knowledge graph is what we call data. It is the set of instance knowledge stored, following the ontology schema and rules. It is defined in the form of inter-connected **entities** (nodes) and **relations** (edges), constructing a graph G. In the graph the entities connected through relations are formally defined as a set of **triples**, being each triple a 3-tuple (h, r, t) where h is the head entity, r the relation and t the tail entity. We can think of a triple as a formal description of a fact. For example, the statement in NL "*Barack Obama was born in Hawaii*" can be expressed with the triple $\langle BarackObama, bornIn, Hawaii \rangle$.

The entities in a knowledge graph correspond to universal semantic representations of word concepts. So, while *words* are dependent from a specific language, *entities* are represented by all words and multilingual aliases referring to the same concept. This kind of representation allows for a) language independence, b) concept universality and c) ontology structure, which allows for constrained representations of relations in the world (as previously described in 0.4.1). Entities allow to discriminate between homonym words and to merge synonymous keywords or aliases under the same entity ID, i.e. the US basketball team "*Golden State Warriors*" is also referred as "*San Francisco Warriors*" or "*Dubs*", so they all represent the same semantics.

In Figure 0.2 we show an example of a knowledge graph structure for the entity "*Marie Skłodowska-Curie*" (Q7186, according to its Wikidata ID). From the example, notice how entities can be of the form of world concepts (e.g. people, organizations, locations, sports, etc.) or literals (e.g. dates, numbers, etc.). Entities are interconnected with triples. However, some relations are only true under some constraints, for example, Marie Curie and Pierre Curie marriage lasted from July 26, 1895 until Pierre death in April 19, 1906. **Qualifiers** are used to express this additional knowledge from triples

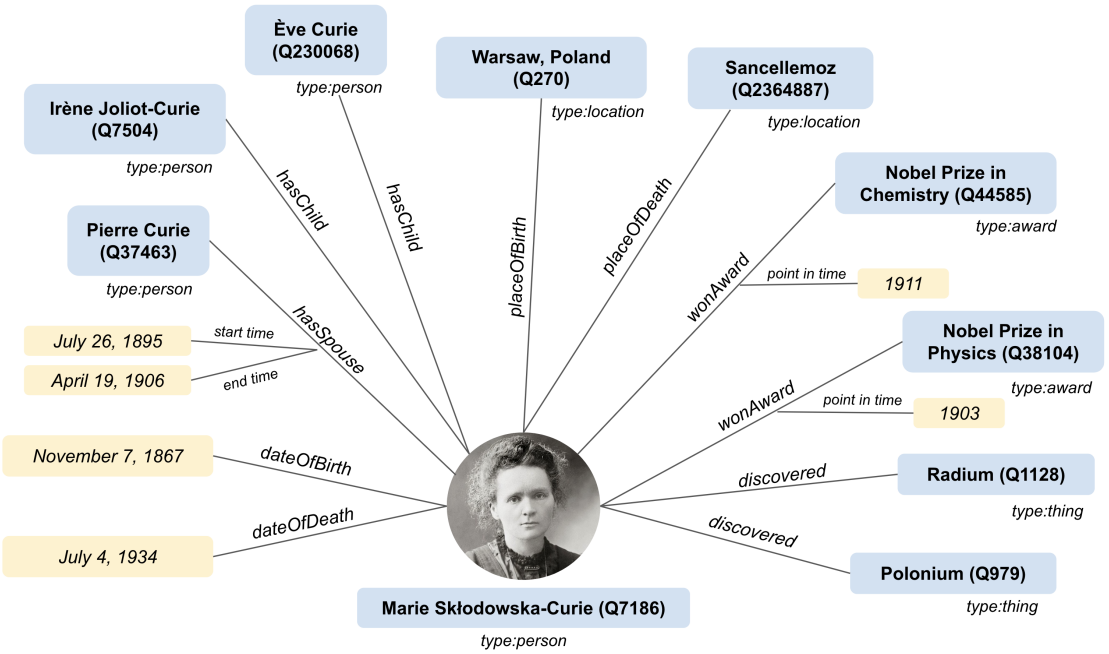


Figure 0.2: Example of a knowledge graph data and relations for the entity of "Marie Skłodowska-Curie" (with Wikidata ID Q7186). Entities are represented in blue boxes and date literals in yellow boxes.

as auxiliary entity-relation pairs. Notice in the image how qualifiers (e.g. "*start time*", "*end time*" or "*point in time*") are not applied to an entity, but to the triple itself. To formally represent this information, an entity representing the triple is created, which is called **statement**.

0.5 Semantic Technologies

The Semantic Web Community has provided several W3C standards, used for the storage, representation, and sharing of information resources in the World Wide Web. In this subsection we will describe the three basic languages used for knowledge graph construction.

- **Resource Description Framework (RDF)**[5]: is a standard graph-based data model, designed for representing and interchanging highly interconnected data on the Web. It represents information in the form of triples or three-part structures consisting of resources, where every resource is identified by an URI (representing abstract "things"), literals (specific data values) and blank nodes. RDF does not support any semantics on its own, other than those carried over from the XML datatype definitions. Representing data in RDF allows information to be easily identified, disambiguated and interconnected by AI systems. Knowledge graphs are usually encoded as $\langle subject, predicate, object \rangle$ RDF triples.
- **Web Ontology Language (OWL)**[6]: is a family of description logic-based languages designed to represent rich and complex knowledge about things, groups of

[5] https://www.w3.org/RDF/
[6] https://www.w3.org/OWL/

things, and relations between things. In knowledge graph construction they are used to define ontologies.

- **Shapes Constraint Language (SHACL)**[7]: is a language for validating RDF graphs against a set of conditions. It is used to maintain data integrity in a knowledge graph by defining the set of rules and constraints that data in the graph must be validated with.

0.6 Public Knowledge Graphs

Different communities and companies invest great efforts on generating their own knowledge graphs. For example, open communities like the Linked Open Data Project (LOD) [8] created DBPedia KG; Metaweb Technologies, Inc. generated Freebase; Open Mind Common Sense generated ConceptNet; and big companies like Google, Facebook or IBM have also defined their own knowledge graphs: Google Knowledge Graph[9], the Google Knowledge Vault [49], the Facebook Graph[10] and IBM Watson's Jeopardy System. Also, some startup companies have specialized on generating knowledge graphs, like GraphAware[11] or UniGraph[12]. Bellow, we briefly summarize the main features of those knowledge graphs which are freely accessible and usable, and thus extendedly used in research:

- **DBpedia**[13]: DBpedia first release was in 2007. It is currently the most popular knowledge graph in the LOD cloud. DBpedia is created from automatically-extracted structured information contained in Wikipedia, such as infobox tables, categorization information, geo-coordinates, and external links. It contains links to many other public knowledge graphs. Main DBpedia is in English, but linked localized versions are available in 125 languages. DBpedia is used extensively in the Semantic Web research community, but also some companies like the BBC [106] or The New York Times [170] use it to organize their contents.

- **Freebase**: its first version was also released at 2007, by Metaweb Technologies, Inc., before being acquired by Google Inc. on 2010. In 2015 Freebase shut down its public services and integrated part of it with Wikidata [147]. Nevertheless, Google Knowledge Graph still allows partially access to Freebase knowledge graph through its public API[14]. In contrast to DBpedia, Freebase provided an interface that allowed end-users to contribute to the knowledge graph by editing structured data. It is multilingual and covers general knowledge.

- **Wikidata**[15]: it is a project of Wikimedia, which started on 2012 as a community effort. Unlike previous knowledge graphs, Wikidata does not only store facts, but also the corresponding sources, so that the validity of facts can be verified. Labels, aliases and descriptions of entities in Wikidata are provided in more than 350

[7] https://www.w3.org/TR/shacl/
[8] https://lod-cloud.net/
[9] https://developers.google.com/knowledge-graph/
[10] https://developers.facebook.com/docs/graph-api
[11] https://graphaware.com/
[12] https://unigraph.io/
[13] https://wiki.dbpedia.org/
[14] https://developers.google.com/knowledge-graph/
[15] https://www.wikidata.org/

languages. It currently integrates equivalent Freebase IDs and other knowledge base identifiers.

- **OpenCyc**: It is part of the Cyc project, which started in 1984 as part of Microelectronics and Computer Technology Corporation. The aim of Cyc is to store (in a machine-processable way) millions of common sense facts such as "Every tree is a plant.". While the focus of Cyc in the first decades was on inference and reasoning, more recent work puts a focus on human-interaction such as building question answering systems based. It is only in English language. Since Cyc is proprietary, a smaller version of the knowledge graph called OpenCyc was released under the open source Apache license.

- **YAGO**: it was developed in 2007 at the Max Planck Institute for Computer Science in Saarbrücken and became publicly available at 2015. YAGO comprises information extracted from the Wikipedia (e.g., categories, redirects, infoboxes), WordNet [58] (e.g., synsets, hyponymy), and GeoNames [202]. All entity names are from English Wikipedia, but can be translated.

0.7 Vilynx Knowledge Graph (VKG)

In this dissertation we will use *Vilynx Knowledge Graph* (VKG) as the core knowledge graph of the technologies developed. This is a commercially-used knowledge graph, operative on the production services of Vilynx. This knowledge graph, contains, thus, the multi-domain and multi-lingual information and semantics needed to represent information from media contents. In this section we will briefly overview its data model (0.7.1) and data (0.7.2).

This thesis contributes to VKG, as it focused on the research, design and data integration for the construction of the first version of VKG. However, the knowledge graph described and used on next sections is more extensive than the initial work, so it is not presented as a main contribution.

0.7.1 Ontology

VKG has an OWL schema inspired by Wikidata. Wikidata was chosen as an inspirational baseline because of its open nature, real-time updates and strong community participation. Equivalently to Wikidata, VKG is structured into *items* and *properties* (also called *relations*). In the familiar terms of semantic technologies, items represent individual entities and classes, and properties resemble RDF properties [51].

The ontology consists on 160 entity-types (or classes) with 21 root-types, and 126 different relations between entities. All classes and relations are mapable to Wikidata classes and properties. Some relations include also qualifiers, which indicate temporal constraints for some properties. Moreover, the schema provides additional properties as *external identifiers* from other knowledge graphs, *descriptions* and multilingual *aliases* (different ways how an entity can be named). Relations in VKG are directed, however triples of symmetric and inverse properties are added during data ingestion. Validation constraints are added to properties and classes using SHACL, with the purpose of maintaining data integrity.

Table 0.1: Example of some of the VKG data stored for the entity "New York". Relations with other entities are excluded in the example. Alias in bold represent the main alias name in each language.

External IDs	Description	Types	Alias
wid: Q60, fid: /m/02_286	City in the United States	Thing, Place, City, AdministrativeArea	**New York City** (en) The Big Apple (en) New York (en) NYC (en) City of New York (en) New Amsterdam (en) **Nueva York** (es) Ciudad de Nueva York (es) **Nova Iorque** (pt) **New York** (tr)

0.7.2 Data

In order to represent media content, the data in VKG contains general encyclopedic knowledge in multiple domains. To fulfil this purpose, this knowledge graph integrates entities from multiple external sources: Freebase, Wikidata, Wikipedia, IMDB and TMDB. Moreover, it ingested several common noun entities from Wordnet [16]. Its base triples are mostly extracted from Wikidata dumps, which also integrate Freebase.

Because of the multilingualism of Vilynx's customers, VKG is a multilingual base, which currently integrates alias of entities in 11 different languages: English, Spanish, Portuguese, Italian, German, French, Catalan, Greek, Dutch, Hindi and Turk. This knowledge base results in a collection of over 3M entities, corresponding to multilingual vocabulary of 16M words, and almost 15M triples, including 9M relations between entities and 5.5M types. Notice that the size of this collection constantly grows when new entities are found on the Internet.

In Table 0.1 we present an example of some of the information saved into VKG for the entity "New York" (Q60). Relations with other entities are excluded in the example. See how for each entity VKG stores external identifiers to other knowledge graphs, the entity description in English, the entity types and entity aliases in multiple languages.

[16] https://wordnet.princeton.edu/

Part I

Information Extraction

Introduction

1

On 2011, IBM Watson [84] was challenged to compete against the two highest ranked players in a nationally televised Jeopardy! game [63]. This game consists on identifying the person, place, thing, or idea described in general knowledge clues, and phrasing each response in the form of a question. The computer showed an impressive capability to understand and answer open-domain questions, beating the human contestants and opening the door to many decision support applications. In order to prepare Watson for Jeopardy!, IBM research scientists programmed it with several algorithms which made it capable of understanding complex natural-language questions and analyzing several sources of information in search for answers. To do that, they faced one of the most currently standing challenges in Artificial Intelligence: how to go from unstructured to structured data representations.

Much of human communication, whether it is in natural language text, speech, or images, is unstructured. In particular, human knowledge generally resides in books, news articles and other unstructured text documents. Nevertheless, computers require data to be expressed in a machine-readable (structured) format in order to exploit the knowledge embedded on it and build intelligent applications, such as semantic search, question answering or machine translation.

Natural Language Processing (NLP) researchers have specialized in solving specific sub-challenges when it comes to understanding the natural language expressed in these documents. In particular, Information Extraction (IE) [172] is an NLP sub-field specialized on the automatic extraction of structured information such as entities, relationships between entities, and attributes describing entities from unstructured sources. The output of IE frameworks are usually triples in the form: $\langle subject, predicate, object \rangle$.

This part of the thesis is dedicated to the problem of extracting structured knowledge from unstructured data, by building IE tools for both entity and relation extraction. This chapter introduces the problems behind unstructured data (1.1), the main IE approaches (1.2) and an overview the contributions of this section (1.4).

1.1 Unstructured to Structured Information

With the raise of the Internet and the enormous proliferation of electronic content, unstructured information (e.g., text, audio, and visual contents) is growing much faster than structured information. Leveraging the knowledge underlying these data is essential for many decision making applications. Nevertheless, extracting this high-valuable knowledge requires the combination of several intelligent systems.

In NLP, understanding the syntax, context, semantics, and different usage patterns, is

needed to infer the meaning of words and phrases. While in structured information, such as traditional databases, the data is well-defined and semantics are explicit, the semantics behind unstructured information are often implicit and must be derived by using background information and inference. Moreover, human language understanding has to deal with *synonyms*, i.e. different words express the same meaning, and *polysemy*, i.e. same words mean different things in different contexts.

Usually, unstructured knowledge residing in text documents is structured in large-scale knowledge graphs which synthesize its semantic information. Encoding text as machine-readable knowledge requires, thus, transforming syntactic constructs to semantic constructs. Such transformation consists on mapping names in text to real word entities and identifying semantic relations among pairs of entities. For example, the word *Apple* may refer to a fruit or to the company *Apple Inc.*, but if found in the sentence "*Steve Jobs was the co-founder of Apple.*", we can infer by the context that it means the company. Moreover, we can extract from the sentence the triple or fact ⟨*Apple Inc.*, *Founded By*, *Steve Jobs*⟩. IE tools cope with these tasks, going all the way from unstructured text to an structured knowledge representation. In particular, entity extraction and relation extraction systems are the base of IE, as will be described later in this chapter.

1.2 Information Extraction Paradigms

When designing IE systems there are two main questions to adress: "which noun phrases are worth extracting facts about?" and, "which are the relevant relations that should be extracted?" When answering these questions, two IE paradigms have emerged, depending on the spectrum of the conceivable answers to these questions: *Closed IE* (or *Ontology-based IE*) [203] and *Open IE* [53] .

1.2.1 Closed IE

In Closed IE, also called Ontology-based IE, the IE process aims to retrieve automatically certain types of information from natural language text. For example, a Closed IE system, would process a set of web pages to extract information regarding geopolitical data (e.g. country, population, capitals, cities, etc.). To guide this extraction process, some kind of model that specifies what to look for is needed. Usually, ontologies serve this purpose by providing a dictionary of entities and its types, and define relations of interest in a specific domain. Ontologies are generally manually-specified by either developers of the ontology or by domain experts.

In these systems there is a coverage limitation in the number of entities and relations that can be populated:

- Regarding entity extraction, closed systems try to recognize entities from knowledge graphs in corpus of text. This is the task of Entity Linking. These models do not cover the detection of emerging entities or missing entities in the knowledge graph.

- In relation extraction, Closed IE approaches treat this task as a classification problem: given a pair of entities co-occurring in a text segment, classify its relation into one of the predefined relation types. Pre-specifying relations of interest is a laborious task, and training such systems requires for large datasets expressing relations

on its different forms. As a result, these systems have traditionally only applied a limited number of relations (of the order of hundreds or less). This leads to sparse knowledge in terms of the relations populated.

1.2.2 Open IE

Open IE is defined as the task of extracting assertions from massive corpora without requiring a pre-specified vocabulary [65]. These systems do not make assumptions about the existence of entities or relations, and are thus ontology independent. Instead, all noun phrases are considered to be entities and the system extracts a large set of relational tuples. The most open systems consider any phrase with a pair of entities to be expressing a relation. Open IE systems have been used to support tasks like acquiring common sense knowledge [121] or learning selectional preferences [161], but offer poor precision for knowledge graph construction.

Even if Open IE offers higher recall than Closed IE, these systems are very sensitive to noise due to the lack of tightly enforced semantics on relations and entities. Also, the fact that every noun phrase in Open IE is a possible entity increases the number of incorrect entities picked in the entity extraction task (e.g. In the sentence "*Two women have been awarded the 2020 Nobel Prize in Chemistry*", "*Two women*" is the subject of the sentence and an Open IE system may create an incorrect entity from it). Moreover, there is no attempt to determine which noun phrases refer to the same entity (co-reference), which difficulties the construction of consistent knowledge graphs.

1.3 Multimodal Information Extraction

Information extraction, has been traditionally defined as extracting information from unstructured or semi-structured text. However, when dealing with news streams, a lot of information is presented in visual or audio forms. Some research has shown that multimodal approaches can provide better accuracy [70, 2]. These systems are based on the premise that information that is difficult to disambiguate correctly in one modality may be easily recognized in another.

Joint multimodal representations have been used for different applications, like visual question answering [5], text-image retrieval [200, 208], image captioning [211], etc. But the use of multimodal data in information extraction tasks has been little explored.

In this dissertation we explore how to integrate information extracted from multimodal sources into an Entity Linking system.

1.4 Contributions

In the next chapters we will present the IE system constructed for Vilynx. It is based in two main parts: **multimodal entity linking** and **relation extraction**.

In Chapter 2 we describe the entity linking system created. This system recognizes named entities from text and combines it with information extracted from multimodal sources in the disambiguation step. We demonstrate how the redundancy provided by non-textual sources helps accomplishing a more accurate disambiguation.

Then, in Chapter 3 we present a closed relation extraction system. This system is capable of recognizing relations expressed between pairs of entities mentioned in a sentence. We demonstrate how state-of-the art relation extraction methods can be enhanced by exploiting semantic information encoded in the knowledge graph.

By concatenating both modules, we construct an end-to-end information extraction framework. This system that extracts triples from text is the main tool needed for knowledge graph population.

Entity Linking

2

To comprehend *what* is being talked about in a text, we need to understand certain information units: e.g. persons, organizations, locations or numeric expressions like time, date and money. These information units are commonly called *named entities* [73]. However, such named entities are only mentions of entities in a text in their surface form (i.e. the form of a word as it appears in the text), and are thus language dependent and not unique. To really understand what a text is about, named entities need to be disambiguated into knowledge graph entities. The task of *Entity Linking* (EL), which is a core sub-tasks of IE, is identifying references to entities in text and mapping them to knowledge graph entities.

The Natural Language Processing (NLP) community has largely investigated the extraction of entities from unstructured text. However, the World Wide Web contains vast quantities of textual information in different formats (e.g. unstructured, semi-structured or tabular), combined with non-textual forms of unstructured data (e.g. image, video or audio). Such data forms were traditionally explored within their own fields, and little research had been done on the intersection of communities. Nevertheless, such multimodal information is often served together in news, blog posts or social media and is thought to contextualize and complement each other. In this dissertation we take an holistic view towards EL by mixing multimodal unstructured and semi-structured sources. We adopt a more robust entity disambiguation by collectively exploiting several inputs and data features.

In this chapter we first describe some background on traditional EL systems and related work (2.1). In the second part we introduce the holistic and multimodal entity linking model that we have developed (2.2), which is able to perform real-time semantic tagging by leveraging multimodal information and combining data from structured, semi-structured and unstructured inputs. Finally we present experiments and model evaluation results (2.3).

2.1 Background

Entity Linking (EL) is the task of recognizing knowledge graph entities in a unstructured or semi-structured text. When dealing with this problem there are two main sub-tasks to be solved: Named Entity Recognition (NER) and Named Entity Disambiguation (NED).

In Figure 2.1, we present an example with the two sub-tasks. First, NER seeks to locate and classify named entities mentioned in a text into pre-defined categories such as person names, organizations, locations, time expressions, quantities, etc. Once mentions are located, NED searches for entity candidates in a knowledge graph and tries to link

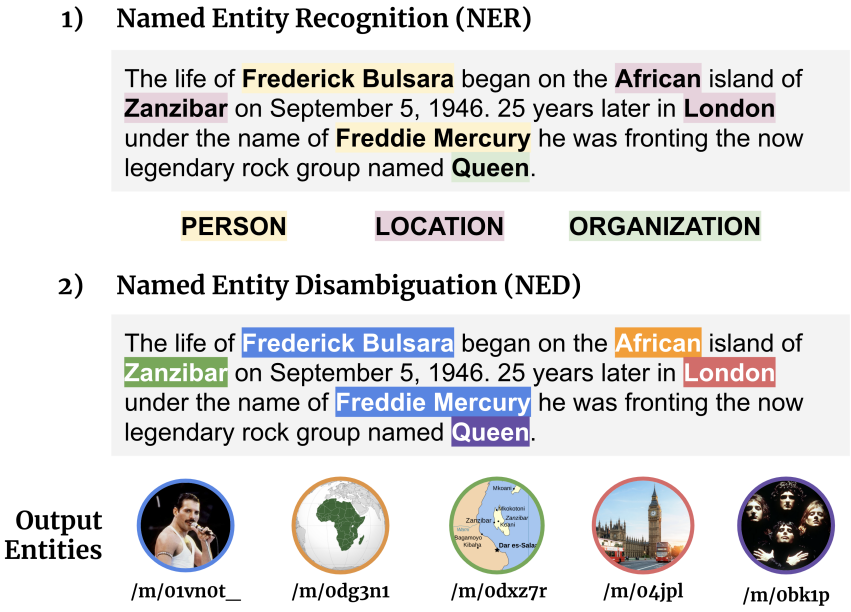


Figure 2.1: Example of a NERD task. 1) NER detects mentions of entities of interest types and 2) NED links such mentions to entities in a KG by solving ambiguities. The output is a set of unique entities (in the example Freebase entities are used).

each mention to an entity. As can be seen, when solving ambiguities, different mentions of named entities, like "*Frederick Bulsara*" and "*Freddie Mercury*", are unified into the same entity, "*/m/01vn0t_*" (Freebase ID). The result of an EL system is a set of unique and universal entities representing real world things.

Most of the state-of-the-art EL approaches are based on textual representations [113, 125, 39]. These approaches have proven their effectiveness on standard EL datasets, such as TAC KBP [96], CoNLL-YAGO [88], and ACE [22]. These datasets contain structured documents with rich context for disambiguation. However, EL becomes more challenging when little textual context is available. On the recent years, some first works on Multimodal Entity Linking have been published. Such works are mostly based on social media [2, 135] and explore methods for adding visual features when disambiguating entities from noisy text with little context, such as tweets. When working with news, we also deal with data with rich multimodal information. News data can be of the form of text, audio or visual (images and videos), as will be studied later in this chapter.

Next in this section we present the two tasks composing an EL model: Named Entity Recognition (2.1.1) and Named Entity Disambiguation (2.1.2)

2.1.1 Named Entity Recognition

The first component in the EL pipeline is responsible for the detection of entity mentions in the document, or named entities. Mention detection is closely related to the problem of NER and a lot of systems perform it with NER techniques. In this section we will focus on this technique for mention detection.

The term "*named entity*" was first used in 1996 at the sixth Message Understanding

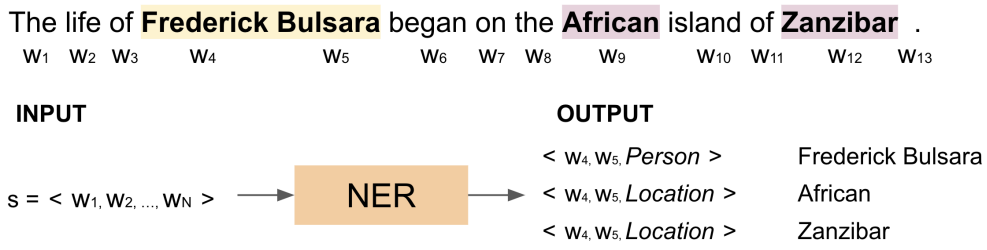


Figure 2.2: An example of a NER task.

Conference (MUC-6) [73]. In such conference, NER was defined as "*the task of identifying names of organizations, people and geographic locations in text, as well as currency, time and percentage expressions*". Since MUC-6, NER has become an important preprocessing step in many fields, like financial journalism [57], biology and bio-medicine [12], and business intelligence [167]. As a result of this diversification, the original range of named entities was progressively extended to include many other entity types depending on the application (e.g., brands, genes, etc.). Nowadays, NER is an essential component in any semantic enrichment or knowledge discovery system, but it also has several applications beyond IE, like machine translation [17] and question answering [134]. Due to its relevance in many fields, various scientific initiatives have devoted much effort to this topic during the past years: e.g. CoNLL03 [171], ACE [47], IREX [43], and TREC Entity Track [19].

2.1.1.1 Task Definition

NER is the process of locating and classifying named entities in text into predefined entity categories. Formally, given a sequence of tokens $s = \langle w_1, w_2, ..., w_N \rangle$, a NER system outputs a list of tuples $\langle I_s, I_e, t \rangle$, each of which is a named entity mentioned in s. Here, $I_s \in [1, N]$ and $I_e \in [1, N]$ are the start and the end indexes of a named entity mention; t is the entity type from a predefined category set. For example, given the sentence in Figure 2.2, the NER system recognizes three entities from the input sentence.

2.1.1.2 Methods

The techniques used in NER have evolved over time. Early NER systems made use of handcrafted rule-based algorithms, while modern systems rely on machine learning and, more recently deep learning techniques. In this section we will briefly describe different NER approaches, and how they have evolved over time, in order to comprehend the present challenges.

Rule-based methods: The first NER systems were designed in the early nineties. Such models relied on refined linguistic knowledge and rules [57]. As such rules had to be encoded manually by linguists, these approaches were extremely time-consuming and costly to maintain. Moreover, such approaches tend to be high on precision but very low on recall, due to the difficulty of manually providing an exhaustive list of all existing possibilities.

Unsupervised Learning methods: Due to the cost of generating hand-crafted rules

and annotated datasets, and the availability of huge corpora, unsupervised learning approaches became a popular solution. In general, these solutions use machine learning techniques like clustering. For example, one can try to gather named entities from clustered groups based on the similarity of context. Other unsupervised approaches use public lexical resources [9], like Wordnet [130], or lexical patterns computed on a large unannotated corpus using statistics [179, 54]. However, totally unsupervised approaches often suffer from a lack of precision, leading researchers to experiment with semi-supervised methods.

Semi-supervised Learning methods: The main semi-supervised technique is bootstrapping, which requires for a set of seeds for starting the learning process [35, 160]. Due to the difficulties and cost associated with finding annotated data, semi-supervised methods raised researchers interests over the years [137]. These methods reached similar performance to supervised ones [139], and have shown to be able to improve traditional supervised methods by using bootstrapping [94].

Feature-based Supervised Learning approaches: Classical machine learning NER systems typically used feature-based supervised methods. Such systems read a large annotated corpus, memorizes lists of entities, and creates disambiguation rules based on discriminative features [138].

Deep Learning methods: Over the past few years, a considerable number of studies have applied deep learning to NER, obtaining the state-of-the-art performance [118]. These methods automatically discover representations from raw input and apply classification and/or detection in an end-to end manner.

2.1.2 Named Entity Disambiguation

NED is the task of mapping words of interest, such as names of persons, locations and companies, from an input text document to corresponding unique entities in a target knowledge graph. In this task, a set of words of interests, called named entities, mentions or surface forms, have already been detected in an unstructured text. The task focuses on the mapping of such words to entities.

Performing NED on real text data is often not trivial, and a robust NED algorithm must deal with a number of different challenges. In Figure2.3 we present an example of NED. In this text we have six mentions, with several possible links to Wikidata entities. Using the named entity types given by the NER module we could discard some of the candidate entities. However, this solution would only partially solve the ambiguity, as there may be many candidates of the same type. To correctly perform NED all entities need to be considered, and - thanks to the context - disambiguated to the correct entities. Following, we list the most common challenges of NED [50]:

- **Name Variants**: entities can appear in different surface forms, because of abbreviations (NY), aliases (Big Apple) or spelling mistakes (New Yokr). Knowledge graph completeness with different entity aliases is essential for dealing with this challenge.

- **Ambiguity**: one of the biggest challenges when solving NED is when the same mention refers to many different entities. This is a common challenge in natural

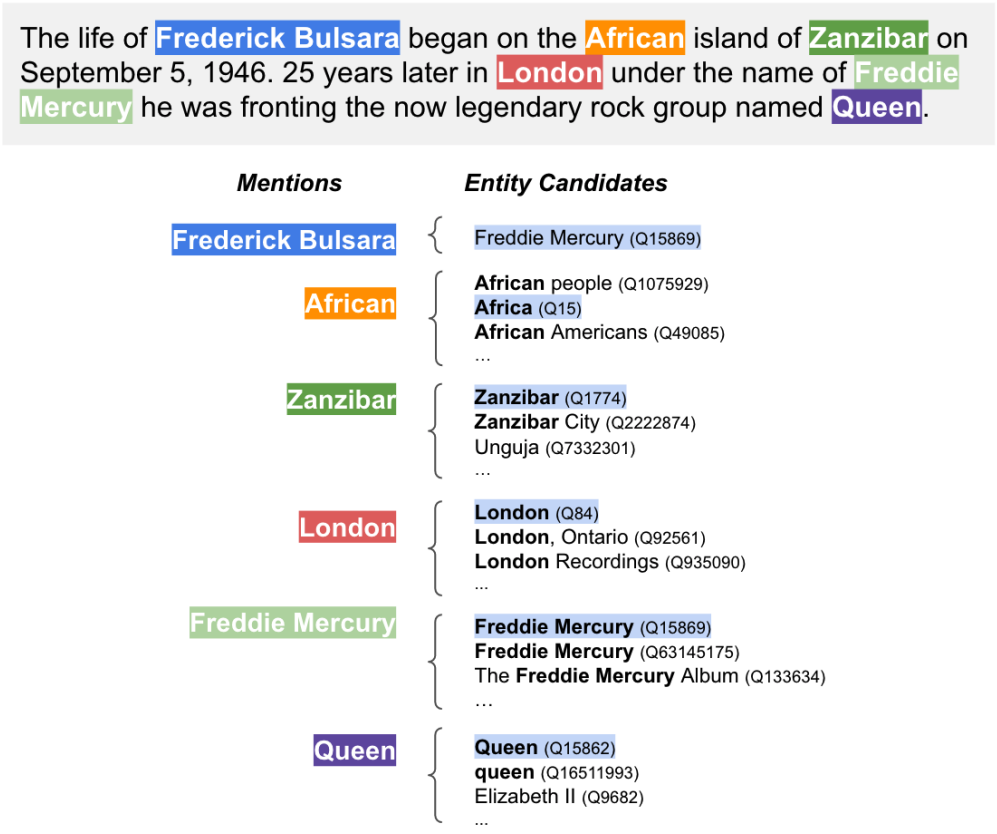


Figure 2.3: An example of a NED task. For each mention detected by the NER system NED chooses the corresponding knowledge graph entity from a set of entity candidates. Correct entities are highlighted in blue.

language, as many entity names are polysemous.

- **Absence**: when processing large text collections, many entities will not appear in the knowledge graph.
- **Incomplete Information**: robust NED systems should be able to correctly disambiguate entities with limited context.
- **Scalability and Speed**: industrial NED systems must provide results in in nearly real-time, which can be challenging when dealing with large-scale knowledge graphs with millions of entities and facts.
- **Training**: in this field, supervised datasets are mostly research oriented and usually contain a few hundred sentences. Therefore, training in a supervised fashion is challenging and most of industrial systems use algorithmic and unsupervised methods as they can provide greater flexibility and often better accuracy.

2.1.2.1 Task Definition

NED is the task of mapping entity mentions from a text to its corresponding knowledge graph entity. Let $M = \{m^1, ..., m^n\}$ denote a sequence of entities mentioned in a text A. The surface form of m_i is denoted by $Name(m_i)$. The named entity type of the entity

m_i is $EntityType(m_i)$. The surrounding context of m_i can be extracted by $Context(m_i)$. Given a knowledge graph containing a set of candidate entities for the mention m_i, being $E_{m_i} = \{e_m^1, e_m^2, ...\}$, the NED task is defined as finding the mapping function $f(m_i)$ that connects each m_i in M to a unique entry e_m^j in E_{m_i}, or the absence of correct entity in E_m for a mention. In this case m_i is labeled as "NIL", whose meaning is unlinkable.

2.1.2.2 EL Methods

To decide which knowledge graph entity best matches the mentions in a text, we first need to retrieve from the knowledge graph the set of entity candidates to map the mention to. Once the candidates are retrieved, the problem consists on selecting the correct entity among the group of entity candidates. While some works approach this as a multi-class classification task[103], the most typical approach is to treat it as a ranking problem [175, 177]. Entity ranking for disambiguation estimates how well a candidate entity matches the context, involving various forms of filtering and scoring of the initial candidate identifiers. An optional step is to determine unlinkable mentions, for which the knowledge graph does not contain a corresponding entity. According to the described steps, typical NED system consists of the following two modules: candidate entity generation, candidate disambiguation.

In the **candidate entity generation** module, for each entity mention $m \in M$ the system returns entities from a knowledge graph (E_m) the mention m may refer to. Approaches to candidate entity generation are mainly based on string similarity between the surface form of the entity mention and the alias of the entity existing in a knowledge graph. It is crucial for the correct entity to be returned inside the candidate entities in order for the NED system to succeed.

Next step in NED, and the most complex and determinant, is **candidate disambiguation**. This consists on selecting a single entity, or none, from the set of candidate entities identified for each mention. Modern disambiguation approaches usually combine two kinds of features: individual and contextual. Individual features are meant to describe the *individual importance* of entities and mentions, while contextual features express the coherence of the entity in a given context. Depending on the context being measured we can differentiate between *contextual similarity* and *entity-relatedness* features.

2.1.2.3 EL Features

Following, we discuss the features used by NED's candidate disambiguation modules. We divide them in the three kinds mentioned in the previous sub-section:

- **Individual Importance**: this kind of features rely on the entity alone or the entity and the mention combined.They try to measure the match between a mention and an entity without taking into account the context. These features can express either similarity between the entity alias and the mention, or popularity.

  When measuring similarity it can either be based on the text mention surface form to the entity alias, or based on its semantic information, like entity types. Typical types of text similarity used are Levenshtein and Hamming distances, or different types of match.

On the other hand, when talking about popularity it can refer either to the entity popularity on its own, or the popularity of a particular entity given a mention. Some popularity metrics to estimate how likely it is that a given text span will be linked to an entity are *keyphraseness* [38], *link probability* [62] and *commonness* [62, 131]. To measure the popularity of the entity itself, some typical and simple estimates are *link prior* [156], which is defined as the fraction of all links in the knowledge repository that are incoming links to the given entity, and *pageviews*[67], which denotes the total number of page views of a Wikipedia entity in a period of time.

- **Contextual Similarity**: these kind of features are based on the premise that information surrounding a mention provides context for disambiguation. These features are computed by comparing the text surrounding a mention with the textual representation (entity description) of the given candidate entity. Both the mention's context and the entity are commonly represented as bag-of-words and the contextual similarity is computed using some similarity function F, being cosine similarity the most commonly used, e.g. [28, 108, 156]. Notice that unlike individual features, which could be estimated independently of the document language, contextual similarity features are language dependant.

- **Entity-relatedness**: these features aim to maximize the entity relatedness between all entities in a document, measuring the overall coherence. It can reasonably be assumed that a document focuses on one or at most a few topics. Consequently, the entities mentioned in a document should be topically related to each other. This topical coherence is captured by developing some measure of relatedness between entities. This relatedness can be extracted from the knowledge graph connections. Another way of measuring relatedness can be based on the co-occurrence of entities in a text corpus (e.g. Wikipedia). In order to operate, entities are projected to some embedding space using knowledge graph embeddings or entity embeddings approaches.

2.1.2.4 EL Approaches

An effective disambiguation system needs to combine local compatibility (which includes individual importance and contextual similarity) and coherence with the other entity linking decisions in the document (entity-relatedness) [18]. This can be expressed in an overall objective function:

$$\Gamma^* = \arg\max_{\Gamma}\Big(\sum_{(m,e)\in\Gamma} \phi(m,e) + \psi(\Gamma)\Big) \tag{2.1}$$

where Γ is a solution set of mention-entity pairs, $\phi(m,e)$ denotes the local compatibility between the mention and the assigned entity, and $\psi(\Gamma)$ is the coherence function for all entity annotations in the same document. Depending on the optimization strategy, we can differentiate between two disambiguation approaches: *individual* and *collective*.

- **Individual disambiguation**: considers mentions individually. This approaches commonly cast the task of entity disambiguation as a ranking problem where each

mention is annotated with the highest scoring entity or NIL if the highest score falls under a threshold.

First EL approaches [38, 28] focused on local compatibility based on contextual features, like similarity and popularity. These approaches would only take into account the first part of equation 2.1, $\phi(m, e)$.

Later approaches improved EL by considering the other mentions in the document, being called individual global disambiguation methods. This idea was introduced by Cucerzan [39], who proposed the assumption of "*entities in a document being correlated and consistent with its main topic*". This work described an assignment of entities to mentions that maximized the similarity between each entity in the assignment and all possible disambiguations of all other mentions in the document. This has the disadvantage of comparing also with incorrect disambiguation, and thus including noisy data. Another strategy introduced later by Milne, consists in first identify a set of unambiguous mentions in the document [131]. The problem with this system is the assumption that there are unambiguous mentions, which generally requires sufficient large documents to be true. Another approach more recently introduced by Ferragina, combines ideas form the two previous methods proposed, is TAGME [62]. This work presents a voting mechanism, where the score for a given mention-entity pair is determined by a collective agreement, like in [39]. This score is based on the commonness and relatedness features presented in [131].

To globally optimize this coherence metrics is an NP-hard problem. However, good approximations can be computed efficiently by considering pairwise interdependencies.

- **Collective disambiguation**: considers all mentions collectively to make a decision. This second approach takes into account semantic coherence across multiple entities in a context. In this case, all entity mentions are disambiguated together, as a disambiguation decision for one entity is affected by decisions made for other entities [175]. The main difference with individual disambiguation is how the maximization of coherence between all decisions is approached.

  First global approach [108] optimizes globally the disambiguation by relaxing it to a linear programming (LP) problem. More recent models use a graph structure [88, 80]. These works represent the connections mention-entity and entity-entity as a weighted undirected graph. The mention-entity edges capture the local compatibility between them, while entity-entity edges represent the coherence. This representation allows for various graph algorithms to be applied.

  More recent works proposed neural architectures. Random walk mechanisms are a common algorithm for optimizing the graph with candidate entities [76, 148]. Globerson et al. [69] reduces the problem by introducing a model with an attention mechanism that takes into account only the sub-graph of the target mention, instead of all the mention candidates in a document. Other techniques approach the global disambiguation with Conditional Random Fields (CRF) [66, 113], and later works [207, 42] studied the application of language models like BERT [44].

Generally, collective disambiguation approaches tend to perform better than individual ones, specially when the document theme is homogeneous. However, the space of possible entity assignments grows combinatorially, in particular for long documents.

2.1.3 Holistic Entity Linking

So far we described traditional EL approaches. These approaches consider text from a single input, but more recent approaches tackle EL with an holistic point of view which may include several inputs. In [143] a first survey with a review of these new approaches is presented. This sub-section will follow parts of this paper, combined with a multimodal EL perspective, to define the concept of holistics and review its applications, as an introduction to our system.

According to Oxford Dictionary, holistic is "*the belief that the parts of something are intimately interconnected and explicable only by reference to the whole*". This concept applied to EL can be understood as EL involving several kinds of inputs, techniques and modalities. For example, the exploitation of distinct multimodal inputs, or the use of diverse NLP tasks for information extraction.

Following, we list the key aspects of holistic EL, according to [143]:

- **Distinct inputs and data features**: Alternative inputs can be used to provide entity descriptions and further knowledge to help EL. However, most EL systems only use text from a single source, but the use of associated metadata, which may include category keywords, locations or timestamps, can boost EL results. For example, [93, 192] use the temporal context of microblog posts to disambiguate, based on temporal entity popularity. Other possible sources are multimodal inputs, which can be either visual or audio. Several works have already shown how using visual data in EL improves disambiguation on social media posts [2, 135]. Using diverse data inputs can boost EL results, specially when limited context information is available.

- **Diverse NLP tasks**: The disambiguation step is usually preceded by the NER task to do the mention detection. However, these tasks have been traditionally considered independent and it has not been until recent years that some papers proposed joint entity recognition and disambiguation methods [66, 125, 181]. Moreover, information from other related NLP tasks can be used to improve disambiguation, like Word Sense Disambiguation (WSD) or Part of Speech (POS) Tagging.

- **Collective Disambiguation methods**: In the definition of holism, it comes implicit the collective understanding of the content. Thus, for an holistic EL system, we expect disambiguation to be performed using global features and an optimization that takes into account all content related information as a whole.

2.2 Method

When willing to perform semantic tagging from news contents, we often encounter that the input is not a single text, but it can include: short text, long text, lists of keywords, categories, dates, images, videos or audio. Such information complement each other and the holistic understanding of it provides a better context of the whole content. To correctly understand and tag such information, we require for entity extraction methods to process information in an holistic and multimodal manner.

Traditional information extraction focused on EL on unstructured text inputs, with a single body of text. Such systems cannot handle multiple format inputs (unstructured and semi-structured) or additional information from multimodal sources (image, video, audio).

In this work we deal with the challenge of creating an EL system for news contents. As previously mentioned, news combine different forms of text, plus multimodal inputs. Thus, such system must be capable of disambiguating entities holistically. The requirements of the system are:

- **Different Text Formats**: we want to create a system capable of extracting entities from news contents on the Web, by processing text from multiple inputs and formats (unstructured and semi-structured). Text may consists on short sentences, like news titles and headline descriptions; long text, like the body of a news article; and lists of keywords or categories found on the metadata. Moreover, multiple URLs may be used to describe a single content.

- **Multimodal Inputs**: contents on the Web have a lot of multimedia contents associated. We want to generate an EL model capable of integrating information extracted from multiple sources: text, image, video and audio.

- **Language Agnostic NED**: disambiguation systems usually depend on the language to generate contextual features. This requires a NED model and datasets for each language, which makes it hard to maintain and train because of the lack of language-based annotated data. Moreover, the EL system created in this work is going to be used for international customers. To facilitate the integration of our technology to new customers we require our models to be as much language independent as possible.

- **Global Context**: we work under the assumption that information from all inputs has a global context. This will require features and scores that measure this overall coherence.

Based on the listed requirements, in the next sub-sections we describe our Holistic EL system. We focus on the fusion of multimodal data and the NED, as our main contributions. Computer vision and speech to text modules are out from the scope of this thesis and we will treat it as black boxes. First, we will overview the whole framework (2.2.1), and will continue describing the two modules we designed for this thesis scope: Mentions Detection (2.2.2) and Holistic Named Entity Disambiguation (2.2.3).

2.2.1 System Overview

Our holistic EL system is designed to tag media documents found on the World Wide Web, e.g. news contents or blog posts, by merging all multimodal information available. In Figure 2.4 we display an schema of the system developed. The documents are indexed with a rich collection of tags associated to knowledge graph entities. The system provides semantic tagging from three different sources: visual, audio and associated text (unstructured and semi-structured). Following, we summarize the behaviour of the main blocks of the pipeline.

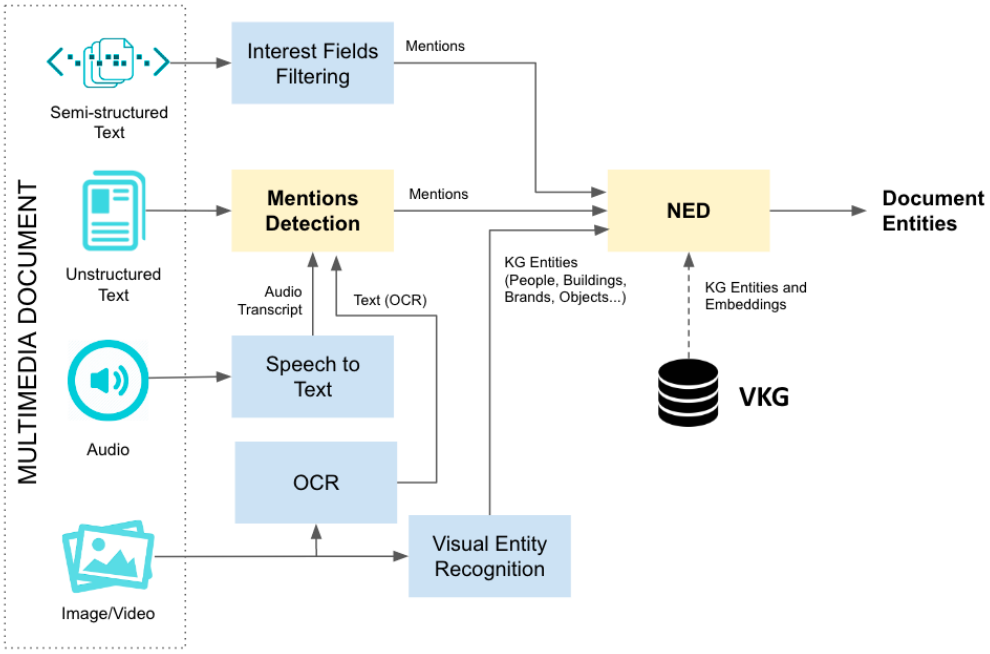


Figure 2.4: Holistic Entity Linking system schema.

1. Document related metadata, like MRSS feed metadata or HTML fields, is processed in the *Interest Fields Filtering* module. Such module selects the content from a predefined set of fields, like *keywords* and *categories*. This textual content is considered a list of named entities of interest that we want to disambiguate.

2. The unstructured text (e.g. title, description, article, post, etc.) is processed in the *Mentions Detection* module. This module combines the output from a NER system with the common nouns extracted from a POS Tagger, to output a set of interest mentions.

3. The audio transcript from videos or podcasts is obtained through speech to text algorithms in the *Speech to Text* module. The entity mentions in the audio are extracted using the *Mentions Detection* module, as for the text sources.

4. The *Optical Character Recognition (OCR)* system detects the text appearing on the images. This text is processed in the *Mentions Detection* module to extract mentions of entities from it.

5. The *Visual Entity Recognition* is composed by many sub-modules and provides detection of the people, places, logos and objects appearing on the video using deep learning analytics. Such modules are trained to detect knowledge graph entities, which will give redundancy and additional context for the textual entity disambiguation.

6. Finally, those mentions extracted from the multi-modal sources are linked to knowledge graph entities by the *Named Entity Disambiguation (NED)* module. Contextual information is enhanced in the collective disambiguation by leveraging information from the detected visual entities.

2.2.2 Mention Detection: NER and POS Tagging

The first step towards the generation of rich tags is retrieving mentions of interest, from all textual unstructured sources available: short text, long text, audio transcripts and OCR text. To do that, the system combines two different NLP techniques: NER and POS Tagging.

As previously introduced in 2.1.2, NER is the task of identifying and categorizing key information (named entities) in text. We will use a NER system to detect general entities such as people, locations and organizations. Nevertheless, some common nouns are also descriptive and of high interest to understand media contents. In order to extract such mentions, we will complement the mentions extracted by the NER with nouns extracted by a POS Tagger. POS Taggers label each word in a sentence with its word class or lexical category (e.g. noun (NN), verb ($VERB$), adjective (ADJ), etc.). We will consider entity mentions the nouns (NN) detected by the POS Tagger.

Both tasks have been long studied by the NLP community, and many libraries incorporate pre-trained models with state-of-the-art performance. Some of these models are even designed for high speed processing, and industrial deployment. As in this part of the thesis our contributions focus on the holistic disambiguation of entities, we decided to incorporate an external library with pre-trained models for these tasks. In particular, we will use Spacy [92] because of its state of the art performance, easy to use Python implementation, multilinguality, and mature adoption in industry.

Finally, the mentions detected from unstructured text sources will be complemented with keywords or categories coming from semi-structured text, like MRSS feeds or HTML metadata. Most EL systems consider each individual mention in a text and its context individually. However, our method aggregates all the same text mentions of entities from different sources or different text occurrences in a unique mention. Such approach assumes the holistic context of all multimodal inputs.

2.2.3 Holistic Named Entity Disambiguation

Classical NED consists on mapping mentions from a text to its corresponding knowledge graph entities. Nevertheless, our holistic entity understanding requires the combination of different entity mentions which can come from sentences of text (with context) or to be isolated keywords. Moreover, the input may also include knowledge graph entities, recognized by the visual modules. Because of this holistic point of view, we will reformulate the NED formal definition as follows:

> Given a set of candidate knowledge graph entities E_{KG} for a set of named entity mentions M, and a set of contextual entities E_c, the EL tasks aims to find the mapping function $f(m_i)$ that maps each mention $m \in M$ to its corresponding entity $e \in E$. In case that the corresponding entity e for the mention m does not exists in E_{KG}, m is labeled as "NIL", which means the mention is unlikable.

Similarly to the classical NED methods, our holistic NED will consist on two steps: *candidate entity generation* and *candidate disambiguation*. First, for each mention, the **candidate entity generation** module queries a knowledge graph for a maximum of K entity candidates. Candidates are retrieved with an Elastic Search query, based on alias similarity and type match with the surface form of the mention. This will reduce the

problem to an small subset of the knowledge graph. Following, the retrieved candidates for each mention will be called a **group of candidates** or **candidates group**.

Once the candidates groups are retrieved for each mention, mentions are mapped to candidate entities in the **candidate disambiguation** step. Disambiguation is approached with an *individual global dissambiguation* approach, treating it as a ranking problem. As described in 2.1.2.4, these scores combine individual features with collective ones in order to select the best mention. Each group of features generates an independent score, which we call *individual score* (2.2.3.1) and *collective score* (2.2.3.2). Both scores assign a probability to the entity (e) - mention (m) pairs, of being the correct disambiguation. Finally, the scores are combined in a final score, $score(e, m)$, for each entity candidate. The combined score for each entity can be expressed as:

$$score(e, m) = \phi(e, m) + \psi(e|\bar{e_1}, ..., \bar{e_N}) \tag{2.2}$$

Where $\phi(e, m)$ is the individual score and $\psi(e|\bar{e_1}, ..., \bar{e_N})$ is the collective score of the entity, given the other selected entities. This score is used to rank the entity candidates of each mention. Finally, mentions are annotated with the most highly scoring entity, or the absence of corresponding entity (NIL) if the highest score falls below a given threshold. The final set of entities selected for each mention can be expressed as:

$$\Gamma^* = \arg\max_{e \in E_m}(score(e, m)) \tag{2.3}$$

Next sub-sections describe how the two scores (individual and collective) are computed, and which features are used in each.

2.2.3.1 Individual Candidate Scoring

Single or individual candidate scoring consists in evaluating the relation between the mention m and each candidate entity e_m^i individually. Namely, we want to compute the probability of the candidate being the correct entity for a given mention, without any extra contextual information. Thus, it consists in finding a scoring function $\phi(e_m^i, m) : E_m \times M \to \mathbb{R}$ such that:

$$\max_i \quad \phi(e_m^i, m) = e_m \quad \forall m \in M \tag{2.4}$$

where e_m is the entity corresponding to the mention.

To compute such score, we formulate it as binary classification problem and keep the probability of being valid as the *individual score*. This means that for each pair of an entity mention m and candidate entity e, a binary classifier is employed to predict the probability of the mapping being *valid* if the candidate entity e is the correct entity for mention m or *invalid* otherwise. It is, thus, a supervised classification problem and any classifier can be used for it.

To feed the classifier we need to compute a feature vector for each entity-mention pair. Following the individual features used are described.

Feature Vector: When constructing the individual feature vector to represent an entity candidate, the most typical features used are similarity features, popularity features and contextual features. Similarity features try to model the similarity between the entity

candidate alias and the mention surface form. Popularity features use prior knowledge on the use of entities and mentions in an external corpus, to provide prior probabilities of the use of an entity. Finally, contextual features, model the similarity between the mention and the entity candidates given the text context where they appear. This last feature is usually computed with word embeddings, making it language dependent, as we would need embeddings for each language we process. On the other hand, these features require mentions with text surrounding it, but in our holistic system we can have mentions coming from list of keywords, which would not have this surrounding context. As we are willing to create a language agnostic and holistic EL system, we do not use contextual similarity features in our feature vector. Thus, the feature vector will be constructed by features that measure the word similarity and the entity popularity. Following, the used individual scoring features are described:

- **Entity Prior**: this feature measures how popular an entity is compared to other entities in a knowledge graph. Given a corpus of documents, it is computed as the number of documents where an entity e is mentioned, divided by the whole number of documents in a corpus. In order to compensate the corpus bias and boost less occurring entities, we compute a logarithmic ratio as in Equation 2.5, where n_e is the number of documents whre the entity e has been tagged, and N the total number of documents. Statistically, more frequently entities are more likely to be the right candidate.
$$P(e) = \frac{\log(n_e + 1)}{\log(N + 1)} \tag{2.5}$$
- **String Similarity**: Evaluates the string similarity between the mention and the aliases of the entity candidate. Common string similarities used for this purpose are Hamming distance or Levenshtein distance.
- **Similarity Ratio**: Other scores evaluating similarity between the mention and entity aliases, use the ratio of common words between both. Inspired by [20], we will create a three positions feature: sample ratio, partial ratio and sort ratio. Sample ratio evaluates the direct ratio of similarity between the most similar alias and the mention. Partial ratio measures if the different words of the alias are included in the mention. And sort ratio measures if the words in the alias appear in the mention but in inverted order.
- **Type Similarity**: Type similarity or type matching is a binary score evaluating whether the extracted category type, extracted by the NER module, matches the entity candidate's types stored in the knowledge graph.
- **Text Search Ranking**: When we query the knowledge graph to get entity candidates, these are returned sorted by a similarity metric. The text search ranking is, thus, the rank position of an entity candidate in the text search query response. We will call it $R_{e_i} = k$, being an integer with values $1 \leq k \leq K$, where K is the maximum number of entity candidates allowed. Previous works have shown this feature provides a lot of information [20].

Classifier: There exist lots of ways of aggregating these partial scores in order to define $\phi(e^i_m, m)$. Most common approaches use weighted averages and ML algorithms, widely covered by the literature. This set of scores aggregated in ϕ will allow to have a good

prediction of the best candidates for each mention. In the experiments section we will evaluate the performance of different binary classifiers to decide which is the best one to use for our use case.

2.2.3.2 Collective Candidate Scoring

Collective Candidate scoring intends to compute if an entity fits the context. In this method, the holistic understanding of all multimodal inputs is translated to the assumption of the existence of a common topic among all the input sources. To this end, all entity candidates of all mentions are considered as a whole, and the score is collectively optimized.

Assuming a common topic or context among all entities, implies that entities in the text should be "related", thus "similar" between them. To model and evaluate this similarity, we will use the distance between entities in an *entity embeddings* space. We require a similarity function $S(e^1, e^2) : E^2 \rightarrow \mathbb{R}$ which inputs two entities and outputs a score about the semantic relationship of the entities. The most typical function to use is the *cosine similarity*, as defined in Equation 2.6, where v_{e_i} is the embedding representation of an entity e_i. According to this metric, similar entities will have a high cosine similarity, whereas totally unrelated entities will have a lower one.

$$S(e_1, e_2) = \cos(\mathbf{v}_{e_1}, \mathbf{v}_{e_2}) = \frac{\mathbf{v}_{e_1} \cdot \mathbf{v}_{e_2}}{||\mathbf{v}_{e_1}|| \cdot ||\mathbf{v}_{e_2}||} \tag{2.6}$$

The output of a collective disambiguation should be the path of entities that maximize the collective scoring function ψ,as expressed in Equation 2.7.

$$(\bar{e_{t_1}}, ..., \bar{e_{t_{|M|}}}) \quad = \underset{(e_1,...,e_{|M|} \in D)}{argmax} \quad \psi(e_1, ..., e_{|M|}) \tag{2.7}$$

All possible candidate path combinations belong to an space $D = E_1 \times ... \times E_{|M|}$. Ideally, since D is a finite set of paths, we could evaluate ψ in all of them and select the best. However, this approach is unfeasible in practice, since the number of tuples grows exponentially with the number of mentions, as expressed in Equation 2.8.

$$|D| = |E_1| \cdot ... \cdot |E_{|M|}| \tag{2.8}$$

In order to approximate such function we use the assumption proposed by [131], which consists on assuming there are unambiguous entities. Based on the knowledge given by the visual entities, as such recognitions are of high accuracy and provide an entity identifier, we construct the initial context and optimize from it. In absence of visually recognized entities, we use entities with very high individual score. Such entities provide a set of disambiguated group of candidates, which we call **reference candidates**. According to the assumption of all entities sharing a common context or topic, we can state that the non-disambiguated entities should be the most related to the reference candidates, i.e. have high similarities with the reference candidates. With this assumption,

we can define $\psi(\mathbf{e})$ as in Equation 2.9, where L the number of reference candidates and $\bar{e_j}$ are the entities corresponding to the reference candidates.

$$\psi(e_1, ..., e_{|M|}) = \sum_{i=1}^{|M-L|} \sum_{j=1}^{|L|} S(e_i, \bar{e_j}) \tag{2.9}$$

Given this approximation, we can compute the collective score of each entity s_c and collectively disambiguate each candidate group as the entity closer to the reference candidates, as expressed in Equation 2.10

$$\bar{e_i} = \underset{e \in E_i}{argmax} \quad \psi(e_i|\bar{e_1}, ..., \bar{e_N}) = \underset{e \in E_i}{argmax} \quad \sum_{j=1}^{|N|} S(e, \bar{e_j}) \tag{2.10}$$

2.3 Experiments

In this section we present the evaluation of the proposed EL system and its applicability as content tagger. To do that, we perform two independent experiments:

The first one measures the quality of the NED model with a dataset (2.3.1). We do not measure the quality of the mentions detection module, because it is mostly based on Spacy library, and our contribution is focused on the NED.

The second experiment (2.3.2) evaluates the quality of the extracted entities as video content tags. This experiment is specially relevant to measure the quality of the system as a commercial product for automatic tagging. For this evaluation we perform a human-based rating through Amazon Mechanical Turk (AMT) crowdsourcing platform.

2.3.1 Model Evaluation

This first experiment measures the performance of the NED module. In this experiment we will analyze features distribution, compare the performance of different classifiers and the effect of adding entities recognized in visual sources.

2.3.1.1 Dataset

The NED model has been trained and evaluated with the VLX-News Dataset. This dataset has been created by Vilynx with the purpose of training and testing the EL model. It consists on a multilingual corpus of 170 news articles from Vilynx's customers. The text from the corpus has been processed by the mentions detection model presented in Section 2.2.2 to extract entity mentions. For each mention, knowledge graph entity candidates have been extracted by querying VKG, until a maximum of 10 candidates per mention. The entity candidates have been annotated by a group of experts that, for each mention, selected the correct entity from the group of candidate entities, or none if any is correct. Final dataset includes sentences with a set of mentions detected with its predicted entity type, and a group of entity candidates for each mention and the label (correct/incorrect) corresponding to each candidate. In Table 2.1 we display the resulting dataset metrics.

Table 2.1: NED dataset metrics. In this table we display the amount of news articles annotated in each language and the number of entities.

Language	N articles	N entities	N correct entities	N incorrect entities
English	128	82312	3400	78912
Portuguese	9	1669	86	1583
Spanish	29	15662	688	14974
French	1	1069	43	1026
Dutch	2	174	6	168
German	1	876	37	839
Total	**170**	**101762**	**4260**	**97502**

2.3.1.2 Features Analysis

To analyze the discriminativity of the features used in the individual scoring classifier, we plotted the density functions of each one of the features for both classes (correct vs incorrect entities). With this analysis we intend to visualize if the chosen features fulfill our assumptions and are thus good discriminators.

- **Entity Prior** (Figure 2.5): correct entities tend to have greater prior values than incorrect ones. However, there are entities that do not have any prior information, specially for incorrect ones, which can produce a bias through this class for unseen entities.
- **Search Ranking** (Figure 2.6): as expected, correct entities tend to appear in highest ranking positions as entity candidates.
- **Similarity** (Figure 2.7): we can see how correct entities have a higher similarity match (similarity = 1), while incorrect ones have a more uniform distribution. Nevertheless, as the candidates are retrieved based on similarity, a lot of incorrect entities have also the highest similarity.
- **Type Match** (Figure 2.8): In this plot we display with 2 the entities that match the type of the mention, with 1 the ones that do not match, and 0 when there is no associated type. Even the knowledge graph is missing a lot of types information, we can conclude that types tend to match for correct entities.
- **Simple Ratio** (Figure 2.9) and **Partial Ratio** (Figure 2.10): both ratios tend to have a value of 1 for correct entities. However, a lot of incorrect entities also have this value, specially for the simple ratio similarity.

2.3.1.3 Training

The NED model described in Section 2.2.3 has two parts that need to be tuned: the individual scoring function and the collective scoring.

For training the individual scoring function we construct feature vectors with the features analyzed in the previous subsection (2.3.1.2). Apart from these features we want to add

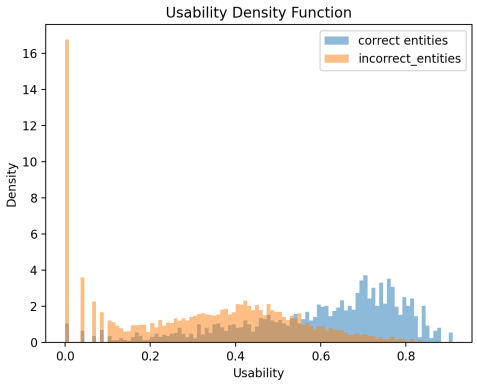


Figure 2.5: Entity Prior

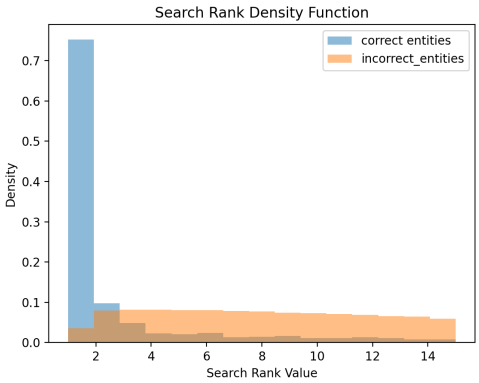


Figure 2.6: Search Rank

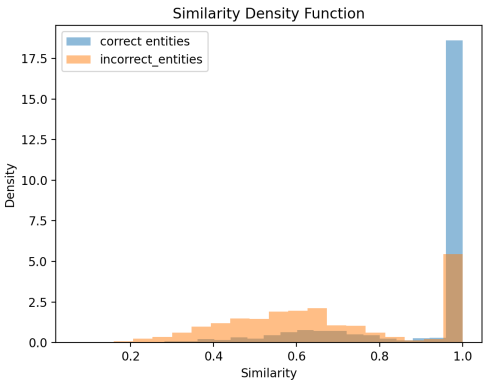


Figure 2.7: String Similarity

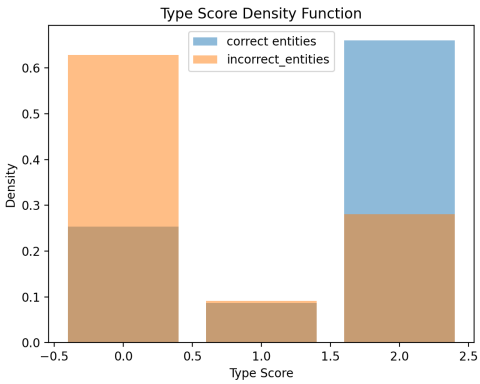


Figure 2.8: Type Match

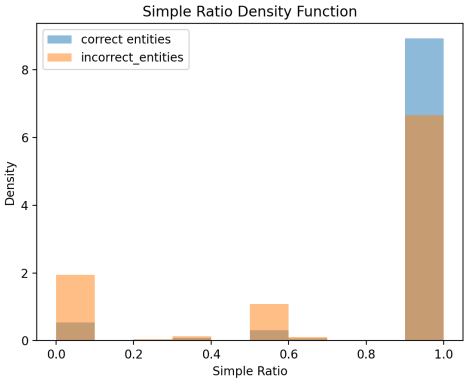


Figure 2.9: Simple Ratio Similarity

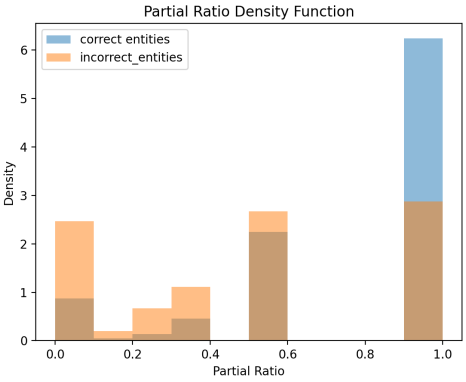


Figure 2.10: Partial Ratio Similarity

Density Distributions of Dataset samples for each feature.

a feature indicating if an entity has been visually recognized in the videos or images related to the news articles. However, VLX-News Dataset is only based on text data. To simulate the multimodality of the corpus we created random mentions of entities with the types recognized by the visual entity recognition models in Vilynx, which are: "*person*", "*building*", "*sport*" and "*brand*". These synthetic visual entities are chosen to match other entities mentioned in the text, with a probability of 0.7, adding redundancy for

these entities.

As individual scoring function of the NED pipeline, we tested different classifiers. These classifiers are traditional ML models: gradient boosting, SVM with differnt kernels (lienar, polynomial and RBF), random forest, bagging classifier and logistic regression. We optimized parameters for each one of them. Neural classifiers are not considered because of the little amount of training samples available.

The second part of the model consists on the collective scoring. For computing it we require an entity embeddings model. We will use Vilynx's entity embeddings, which is a proprietary model trained with entity co-occurrences in a news corpus. As opposed to the majority of knowledge graph embeddings, entity embeddings capture the semantic similarity between entities depending on their relative distance. The objective for this kind of embeddings is that the representation of semantically related entities are closer in distance than unrelated entities. However, only 68% of the entities in the dataset have an embedding, thus, entities without embedding will not have contextual score.

Finally, the dataset is partitioned into a train/test sets where 70% of the dataset articles are set to train and the remaining 30% for test.

2.3.1.4 Results

The NED results are presented in Table 2.2. In this table we compare the overall system performance with different classifiers and adding the visual entity feature vs not adding it. Metrics displayed are computed setting a threshold of 0.5 to the final score. SVM classifier with Polynomial kernel is the one showing the best results, followed by SVM with RBF kernel, reaching an f1-score of 0.82 and an accuracy of 0.74 when using multimodal information. Notice in the results table how adding the information of visually recognized entities increases a 5% th final f1-score for the SVM Polynomial and the accuracy improves a 6%.

We also studied the effect of final score thresholds. This threshold has been optimized through precision-recall curve. In 2.11 we display the resulting curves.

2.3.2 Tagging Quality Evaluation

The entities extracted by the EL system presented in Section 2.2 are used as tags to describe the contents. Thus, in this context, we will refer to the entities associated to the content as tags. These tags have relevant content metadata which is leveraged by Vilynx's search and recommendation products. In order to evaluate the quality of the tags extracted with the presented EL system and its usability as content descriptors, we designed a rating task. This experiment also intends to evaluate if video associated metadata can be used for leveraging relevant entities that describe video contents. Thus, this experiment does not extract entities from visual nor audio data, only uses the associated text and metadata.

The quality of the tags generated by the EL system proposed is assessed on a subset of videos from the YouTube-8M Dataset [1]. The resulting entities are evaluated by human raters from the Amazon Mechanical Turk (AMT) [144] crowdsourcing platform.

This sub-section describes the contents of the video subset used in the experiment, the

Table 2.2: NED results comparison. For each of the tested classifiers we compare its performance in terms of precision, recall, f1-score and accuracy.

	Classifier	Precision	Recall	F1-Score	Accuracy
Traditional EL	SVM (Poly)	0.70	0.84	0.77	0.68
	SVM (RBF)	0.68	0.86	0.76	0.67
	SVM (Linear)	0.70	0.75	0.72	0.64
	Random Forest	0.70	0.82	0.76	0.67
	Bagging	0.69	0.83	0.76	0.67
	Gradient Boosting	0.69	0.81	0.75	0.66
	Logistic Regression	0.59	0.98	0.73	0.60
Multimodal EL	SVM (Poly)	0.76	0.89	**0.82**	**0.74**
	SVM (RBF)	0.74	0.91	0.81	0.73
	SVM (Linear)	**0.78**	0.83	0.80	0.73
	Random Forest	0.77	0.84	0.80	0.73
	Bagging	0.77	0.85	0.81	**0.74**
	Gradient Boosting	0.69	0.81	0.75	0.66
	Logistic Regression	0.64	**0.99**	0.77	0.66

statistics of the generated tags, and the assessment of their quality with AMT.

2.3.2.1 Dataset

Our experiments use a subset of 13,951 videos from the public YouTube-8M video dataset [1], each of them annotated with one or more tags. Given the URL from each video, we parse its contents to extract its textual unstructured and semi-structured information (title, description and metadata). This text information may include different languages, given the multilingual nature of the YouTube-8M dataset. Moreover, YouTube-8M labels are Freebase entities, which allows a comparison between the original tags and the enhanced tags that our system provides.

The 13,951 videos from the subset were randomly selected and cover a large vocabulary with a wide number of topics. Figure 2.12(a) shows the distribution of videos included in the subset for the top-20 most repeated entities. Notice how the subset has a bias towards video games, vehicles, sports and music related entities, a distribution similar to the full YouTube-8M dataset.

2.3.2.2 Tagging Statistics

A total of 34.358 distinct knowledge graph entities were extracted by the EL system from the 14k videos. In Figure 2.12(b) we show the top-20 most repeated tags extracted by our system, compared to YouTube-8M's in Figure 2.12(a). Notice a similarity on the top-level categories of the concepts: "Music", "Vehicles", "Video Games", "Food" and "Sports".

The average number of entities per video extracted is 10.04, while the average number of entities in YouTube-8M dataset for the same subset of videos is 3.64. Nevertheless, in YouTube-8M tags have gone through a vocabulary construction, where all entities must

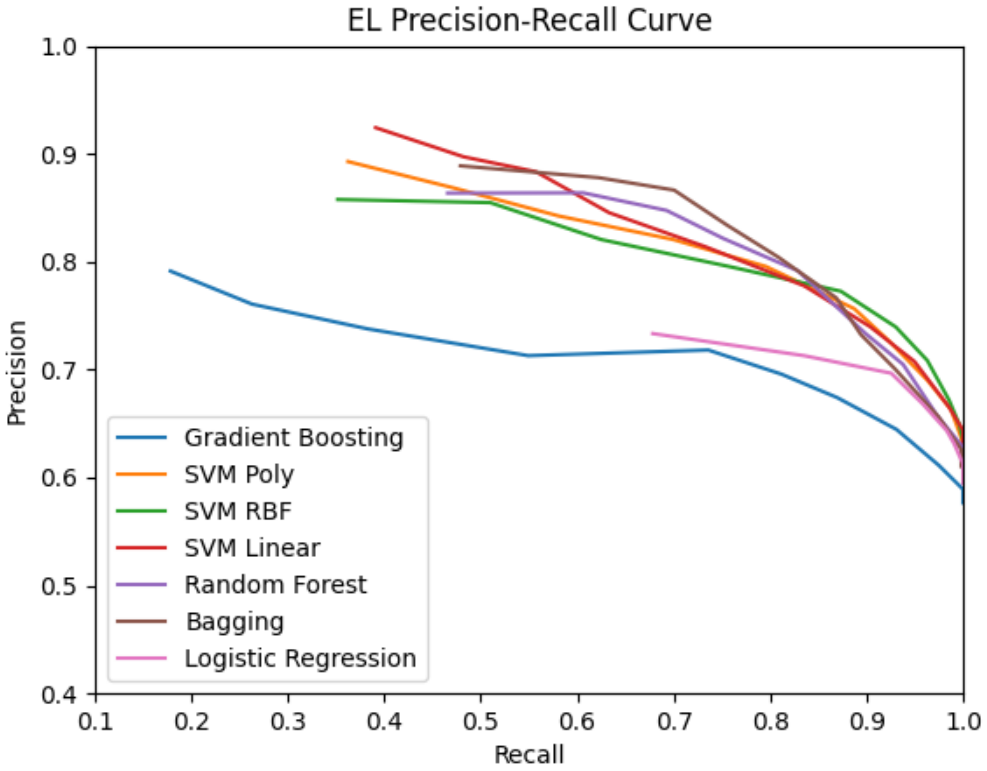
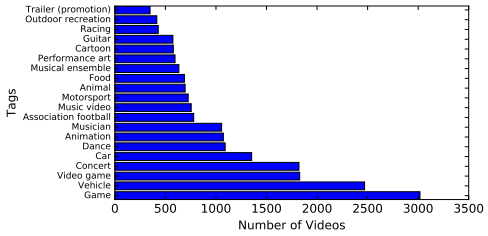
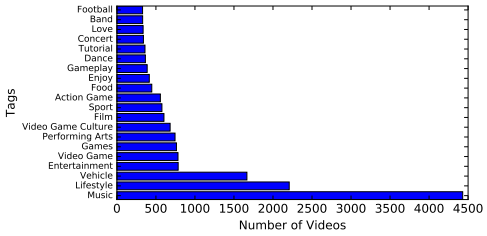


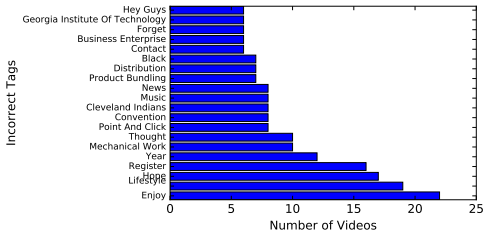
Figure 2.11: Precision Recall Curve for the tested classifiers when varying the total score threshold from 0.0 to 1.0 in 0.1 steps.



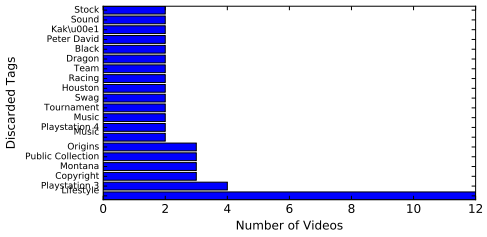
(a) The top-20 most repeated tags in the YouTube-8M subset.



(b) The top-20 most repeated tags extracted.



(c) The top-20 tags evaluated as incorrect in more videos.



(d) The top-20 tags that have been discarded in more videos.

Figure 2.12: Entities statistics from the AMT results.

Table 2.3: Multilingual Tagging Statistics

Language	**#Videos**	**Average #Tags**
en	6,806	12.11
null	5,297	8.83
es	450	5.99
de	246	6.53
it	227	6.39
id	140	6.54
pt	135	4.54
nl	104	8.15
fr	90	5.68
ca	52	5.15
ro	49	6.83
tl	42	4.02
af	34	5.58
hr	30	6.06
no	28	5.92
Total	**13,951**	**10.04**

have at least 200 videos in the dataset, and also only entities with visual representation are allowed, as described in [1]. In Table 2.4 we show a comparison of our extracted entities with respect to YouTube-8M ground truth tags for three videos. Notice the specificity and the higher quantity of entities our system provides.

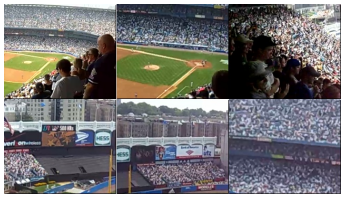
Table 2.3 contains the average number of entities extracted depending on the language of the contextual information. Language is recognized by using a Wikipedia based language detection algorithm [180]. When we do not recognize the language (*null* in the table), we treat it as English. Notice how, due to the fact that most of the videos in the subset are in English, this produces a bias in the knowledge graph vocabulary, which is larger for English aliases. Also, relations of English topics are better learned than others. As a consequence, the average number of entities per video is higher when the contextual information is in English.

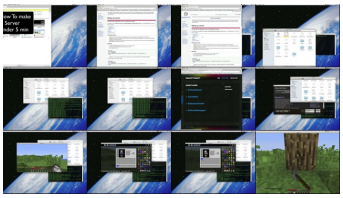
2.3.2.3 Human Rating of Generated Tags

The automatic annotations from the contextual information can be noisy and incomplete, as it is automatically generated from video title, description, metadata and user comments on social networks. The quality of the automatically generated entities was assessed by human workers from the Amazon Mechanical Turk (AMT) online platform. The entities from 1.4k randomly selected videos were shown to AMT workers, limiting the experiment to videos in English and workers located in the United States.

In each HIT (Human Intelligent Task) from AMT, three different workers evaluated the correctness of at most 10 entities assigned to the video, ranked according to a Vilynx's relevance scoring algorithm. If the video had more than 10 entities associated, the additional entities were not evaluated. The video summaries, title and description from the video were shown to the worker on the user interface depicted in Figure 2.13. Workers

Table 2.4: Comparison between Vilynx and YouTube-8M Tagging





Vilynx	YouTube-8M	Vilynx	YouTube-8M	Vilynx	YouTube-8M
Baseball	Game	Thomas Robinson	Basketball	Minecraft	Game
Alex Rodriguez	Arena	Sacramento Kings		Video game	Minecraft
New York Yankees	Athlete	New Jersey		Server	
New York City	Baseball park	Sport		Browser extension	
Yankee Stadium	Stadium	2012 NBA Draft		Tutorial	
SportHit	Home run			Download	
Home run				Video game culture	

Title: "Pet Rescue Saga Level 627"

Description: "Pet Rescue Saga Level 627 played by http://www.skillgaming.de Pet Rescue Saga Walkthrough Playlist: http://bit.ly/18D3UXq Pet Rescue Saga Facebook: http://apps.facebook.com/petrescuesaga/"

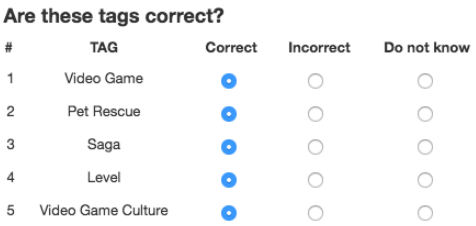


Figure 2.13: Example of AMT HIT layout. On the left, video summaries are displayed in loop, together with title and video description below. On the right, the extracted tags for the video are shown for their evaluation with radio buttons.

were asked to decide if the entity were correct based on that information. For each entity, the worker was asked to select one of these options: *Correct, Incorrect, Do not know.* The '*Do not know*' option was added because entities may be sometimes very specific and difficult to recognize by a non-expert rater, but should not be considered incorrect for this reason. An answer was accepted when at least two workers agreed on it. If all three workers voted for the same option, we refer to it as 'absolute correct'. In case of complete disagreement, or if workers vote for majority the '*Do not know*' option, the entity is discarded. Entities extracted by our EL system that also appear in YouTube-8M ground truth were considered 'absolute correct'. Thus, these entities were not shown to the workers, but are accounted in the provided results.

Table 2.5 provides the accuracy results. We obtained a correctness of 77.81% of the entities evaluated, with a 77.31% of this entities with 'absolute correctness' (agreement of all 3 human raters or already in YouTube-8M annotations). Note that typical inter-rater

Table 2.5: Tag Quality Evaluation Results

#Videos	#Tags Total	Accuracy
1,400	14,024	80.87

% Correct	% Incorrect	% Discarded
77.81%	18.27%	3.90%

agreement on similar annotation tasks with human raters is also around 80% [14, 145], so the accuracy of these labels is comparable to (non-expert) human-provided labels.

We also analyzed the most repeated errors and uncertain tags. Figure 2.12 shows the top-20 entities with more occurrences, evaluated as incorrect or discarded. Notice that many of these entities are too generic concepts, such as 'Lifestyle' or 'Music', which are often found on automatically generated metadata. Also, most of the incorrect entities are abstract concepts, like 'Enjoy', 'Hope', 'Year' or 'Thought', that are often found on contextual information but are not descriptive nor relevant to the video. Moreover, we found some incorrect entities caused by repeated errors on the mapping from keywords to knowledge graph entities, such as 'Georgia Institute of Technology' coming from the keyword 'technology', 'Trip Tucker' coming from 'trip' or 'Head of Mission' coming from 'cmd' or 'com'.

2.4 Conclusions

In this chapter we presented an holistic EL system. Holistic approaches have the potential to boost EL by exploiting several data features and processing methods to make the highest possible number of semantically coherent links. The purposed system includes many novelties, as the combination of multimodal inputs and different text formats for a more robust and collective understanding of contents. Moreover, as stated in the system requirements, we created a language independent NED model that focuses only on the language dependencies to the mention detection module.

The EL system was evaluated with two experiments. The first experiment had the purpose of getting an objective metric of the system performance and evaluate how visually recognized entities improve overall system performance. It was evaluated with VLX-News Dataset, which was used for both training and testing. The performance of different ML classifiers was evaluating, getting an overall f1-score of 0.77 by the best performing system (SVM Polynomial). This performance was improved by a 5% (reaching 0.82) when simulating the addition of visually recognized entities.

The second experiment assessed the quality of the extracted entities as video content tags. The EL system was tested on a subset of videos from the YouTube-8M dataset. The tags generated were highly graded by human users exposed to a visual summary of the video and its metadata. The accuracy of 80.87% is comparable to the inter-annotator agreement of (non-expert) humans in the task of semantic annotation. This high quality, combined with its capability of capturing not-only visual concepts, shows its capabilities as a rich multimodal indexing system.

The presented EL system shows how an holistic understanding of contents can improve

EL. The system described is also an industrial tool, which proved its accuracy and performance with different media customers and tags thousands of media contents every day.

Relation Extraction

3

The goal of Information Extraction (IE) is to extract specific kinds of information from text documents and output them in a structured manner. In particular, named entities and relations. Relation Extraction (RE) is thus the second step in an IE system, after Entity Linking (EL), and is an indispensable task for answering user queries from unstructured text more effectively. The task of RE focuses on understanding how entities co-occurring in the same text segment relate to each other. When willing to recognize relations predefined in an ontology, the RE task can be understood as a classification problem.

In this chapter we will focus on studding closed, or ontology-based, RE methods. We will start overviewing background knowledge related to this task (3.1), and continue by presenting our contributions for enhancing state-of-the-art models by leveraging semantic knowledge (3.2). Finally, we will present our conclusions (3.4).

3.1 Background

Relation Extraction (RE) [219, 81] is the task that seeks to extract semantic relations between the detected entities in the text. In this section we will formally define this problem (3.1.1), and overview the distant supervision concept (3.1.2) and methods used to solve this task (3.1.3).

3.1.1 Task definition

In the RE task, we want to learn mappings from candidate facts to relation types $r \in R$, where R is a fixed dictionary of relation types. We add the no-relation category, to denote lack of relation between the entities in the candidate fact. In our particular

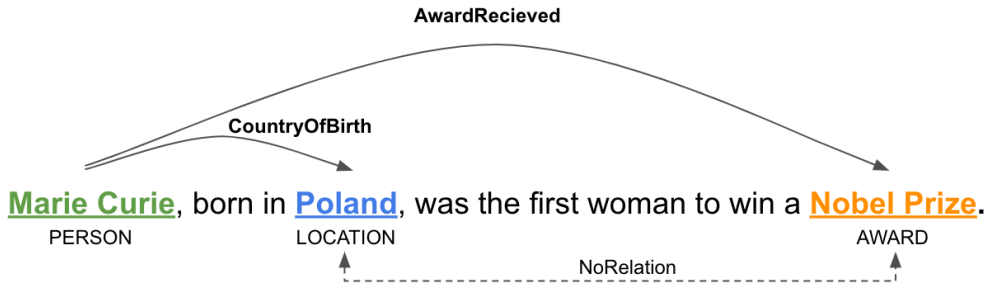


Figure 3.1: An illustration of a RE, where name entities have been recognized. The RE tasks consists on extracting the relations or the absence of relations between entities.

implementation, a candidate fact $(\mathbf{x}, \mathbf{e}_1, \mathbf{e}_2)$ is composed by a set of tokens $\mathbf{x} = [x_0 ... x_n]$ from a sentence s, with a pair of entity mentions located at $\mathbf{e}_1 = (i, j)$ and $\mathbf{e}_2 = (k, l)$, being pairs of integers such that $0 < i \leq j$, $j < n$, $k \leq l$ and $l < n$. Start and end markers, $x_0 = [CLS]$ and $x_n = [SEP]$ respectively, are added to indicate the beginning and end of the sentence tokens. Our goal is, thus, to learn a function $r = f(\mathbf{x}, \mathbf{e}_1, \mathbf{e}_2)$ that maps the candidate fact to the relation type expressed in $\mathbf{x}$ between the entities marked by $\mathbf{e}_1$ and $\mathbf{e}_2$.

In Figure 3.1 we present an illustration of a RE task. In this example, candidate facts would be constructed for all pairs of entities extracted, taking into account both relational directions. In this particular example, the system would generate six candidate facts between each possible pair of the three detected entities. Then, a classifier would predict the relation or absence of relation between the entities in the tokenized sentence $\mathbf{x}$. Notice that as relations are directional, the candidate fact $(\mathbf{x}, MarieCurie, Poland)$ would resolve on the relation *CountryOfBirth*, but the candidate fact $(\mathbf{x}, Poland, MarieCurie)$ would resolve to *noRelation*.

3.1.2 Distant Supervision

One of the main problems in RE, is the lack of annotated datasets. Moreover, the publicly available datasets are mostly in English language and only from specific domains, e.g. news, bibliography or medicine. Such limitation prevents RE methods from scaling both in the amount of relations, languages, and applicable domains. The distant supervision [132] paradigm was presented as a method for automatically generating large RE datasets, by stating:

> *"Any sentence that contains a pair of entities that participate in a known [...] relation is likely to express that relation in some way."*

Based on this hypothesis distantly supervised datasets can be obtained by aligning knowledge graphs with large text corpora that contain annotated entities.

Ridel et al. [159] noted that the distant supervision hypothesis was not true in many cases. Namely, a sentence containing two entities that share a relation does not necessarily express that relation. However, they did assume that if two entities share a relation, in the whole corpus, at least one sentence in which the entities appear has to express the relation. Therefore they relaxed the distant supervision hypothesis by introducing the *expressed-at-least-once* assumption:

> *"If two entities participate in a relation, at least one sentence that mentions these two entities might express that relation."*

The resulting dataset then consists in bags of sentences that contain pairs of entities which are related with one another. Up until this point, all models made sentence-level predictions. With the introduction of the new distant supervision hypothesis, bag-level models were needed. This models make a single relation prediction for the whole bag of sentences where an entity pair has appeared. From this point onward, we can distinguish two kind of models, depending on the dataset form: *sentence-level* or *bag-level*. Models that trained with supervised datasets or distant datasets created with the original hypothesis are *sentence-level*, whilst models trained with distant datasets created with the relaxed hypothesis from [159] are *bag-level*.

3.1.3 Methods

RE methods have evolved over time from using hand-crafted and bootstrapped rules and patterns, to sophisticated deep learning methods. In this section we will overview such approaches.

3.1.3.1 Semi-supervised

Initial research efforts in RE focused on applying manually crafted or automatically extracted linguistic rules and patterns. Most of these methods are semi-supervised and rely on bootstrap learning. Semi-supervised methods consists on using a small dataset to learn patterns with which to extend the initial dataset, and then repeating this procedure in an iterative fashion. In general, these first methods had a decent precision, but the recall was limited to relational patterns.

One of the first and most popular semi-supervised RE methods is DIRPE [27], which presents the *Pattern Relation Duality* algorithm to extract triples from an initial set of seed samples in an iterative way. Snowball [4] extends DIRPE by adding named entity types and computing a confidence over the extracted patterns, in order to reduce false positives. Lately, KnowItAll [54] presented an autonomous, domain-independent system that extracts facts from the Web. Unlike previous systems, KnowItAll uses only generic hand written patterns based on general noun phrases. URES [162] presents an extension on top of KnowItAll to extract instances of relations from the Web by taking as input the definition of the target relations. More recently, TextRunner[210] introduced a novel approach which discovers relations automatically, without using any seed of predefined relation types.

3.1.3.2 Supervised Machine Learning

In the supervised domain, the RE and classification tasks specifically refer to the classification of an entity pair to a set of known relations, using documents containing mentions of the entity pair [109]. Traditional machine learning methods for RE use annotated datasets to optimize classical machine learning models like Support Vector Machines (SVM) [77], voted perceptrons, Maximum Entropy (ME) classifiers [102], etc. Unlike the pattern based algorithms, which predicted a single relation for each pattern, machine learning methods are defined as multi-class classification problems, therefore training a single model for all relations is enough.

Machine Learning methods can be divided in two groups: *kernel based* [212, 40, 29, 136] or *feature based* [77, 102, 98]. Kernel methods use the kernel function during training to evaluate the similarity between sentences, while feature based methods need to precompute a set of features for comparison. Feature based methods integrate different kinds of features extracted from the sentences (e.g. lexical, syntactic and semantic features), in order to create representations which can be input into a classifier.

3.1.3.3 Supervised Deep Learning

With advances in deep learning and the increase in computational power, deep learning models showed to surpass classical machine learning methods when large amounts of annotated data is available.

The early works using deep learning for RE worked in training supervised models with hand-annotated corpus [109]. In 2013 the first use of Convolutional Neural Networks (CNN) for RE [124] was presented. This work tried to use a CNN to automatically learn features instead of using hand-craft features. It was closely followed by [214], which introduced the use of pre-trained word embeddings and a max-pooling layer after the convolutional one, to capture most useful features. Lately, [141] was built upon these previous models, incorporating convolutional kernels of varying window sizes to capture wider ranges of n-gram features.

On the following years, researchers started studying how to exploit the large training data created by distant supervision [132, 159] while being robust to the noise in the labels. This was done by modeling the task as a multi-instance learning problem. Multi-instance learning is a form of supervised learning where a label is given to a bag of instances, rather than a single instance. In RE, an instance is equivalent to a tokenized sentence with annotated entities. Piecewise Convolutional Neural Networks (PCNN) model [213] presented a multi-instance learning paradigm, with a neural network model to build a RE using bag-level distant supervision data. However, this model only used the most-relevant sentence from the bag, loosing a lot of useful data. To address this shortcoming, [123] used an attention mechanism over all the sentences in the bag. This mechanism led to the effective use of all the informative sentences whilst reducing the influence of wrongly labeled ones and showed improvements when using either CNNs or PCNNs. Another way of addressing the problem of information loss in [213] is through the use of multi-instance multi-label CNNs as in [99].

More recent works proposed methods for jointly extracting entities and relations. CoType [157] computes embeddings for entities and relations, and projects new sentences into the latent space where the corresponding entities and relations are found the closest. Another work [224] proposed a combined architecture with a BiLSTM encoder-decoder for entity extraction and a CNN for relation classification.

Another recent approach [154] incorporates the use of reinforcement learning to generate a false-positive indicator to perform hard decisions on the validity of sentences, avoiding both selecting a one-best sentence and calculating soft attention weights over bags of sentences.

3.1.3.4 Supervised Deep Transformers

With the introduction of the Transformer architecture [195] many powerful language models arose, like BERT [44], Transformer-XL [41] and GPT [155]. Given a text corpus, a Language Model (LM) is a probability distribution over sequences of words from the corpus vocabulary. Namely, a function able to predict the probability of any possible sentence as a combination of the words that appeared in the corpus [34].

Language models can be used as a base to transfer existing knowledge of a model trained for a specific task to another similar or related task. This technique is commonly used in deep learning in order to save plenty computational power. In RE, as well as most NLP tasks, new state-of-the-art was established by adding simple architectures on top of pre-trained language models.

The most commonly used language model in RE studies is BERT [44]. It is an unsuper-

vised transformer trained for two tasks: masked language modeling and next sentence prediction (i.e. predict if a chosen next sentence was probable or not given the first sentence). BERT's model captures the contextual information of a word in a given sentence, along with the semantic relation of a sentence to the neighboring sentences in building the whole text [16].

One of the first language model adaptations for RE [178]. In this work a BiLSTM and a fully connected one-hidden-layer is build on top of BERT to adapt it for RE. During the training, they replace entity mentions by masks indicating subject/object function and type, to avoid overfitting. R-BERT [204] presents an architecture that uses markers to indicate entity spans in the input and incorporates a neural architecture on top of BERT to add information from the target entities. A similar input configuration is presented in Soares et al. [183], by using *Entity Markers*. Moreover, they test different output configurations and obtain state-of-the-art results when training with *Matching the Blanks* (MTB) method. Inspired by these previous works, SpanBERT [100] has been proposed as an extension of BERT that uses a pre-training configuration which masks spans instead of tokens. Other works like ERNIE [221], KG-BERT [209] or KnowBert [149] propose enhanced language representations by incorporating external knowledge graphs.

3.2 Method

As previously defined, RE is the task of predicting the relations or properties expressed between two entities, directly from the text. Semantics define different types of entities and how these may relate to each other. Previous works [164, 31] have already shown that entity-type information is useful for constraining the possible categories of a relation. For instance, family-related relations like *Parents* or *Siblings* can only occur between entities of type *Person*, while *Residence* relation must occur between entities of type *Person* and a *Location*. Recent advances in NLP have shown strong improvements on RE when using deep models, specially deep transformers [195]. In this section, we explore different input configurations for adding entity-type information when predicting relations with BERT [44], a pre-trained deep transformer model which is currently giving state-of-the-art results when adapted for RE. The remainder of the section starts by introducing *Type Markers* (TM) (3.2.1), our novel proposal to encode the root type of the entities. We follow by presenting the different input model configurations proposed to add *Type Markers* (3.2.2). And we finish the section by presenting the experimental results (3.3) on the proposed models.

3.2.1 Introducing Type Markers

In this thesis, we present the novel concept of *Type Markers*, which are used to add entity-type background knowledge into the RE model. These markers are special tokens representing the root type of an entity, e.g. [PERSON], [LOCATION], [ORGANIZATION], [WORK], etc. These new tokens are added into BERT embeddings, and their representation will be learned when fine-tuning our model. For each entity in a candidate fact, its type can be extracted from the knowledge graph. However, as knowledge graphs are often incomplete, type information may be missing for some entities. In this case, the entity-type extracted by a Named Entity Recognition (NER) [112, 138] system can be used. In the next section we propose two methods to include these tokens into the

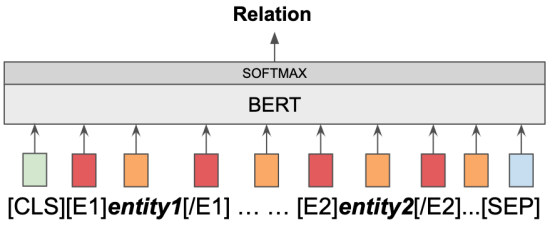


Figure 3.2: Entity Markers[183]

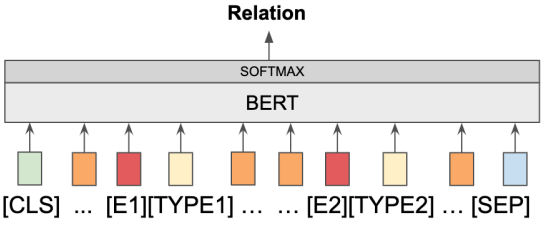


Figure 3.3: Type Markers only

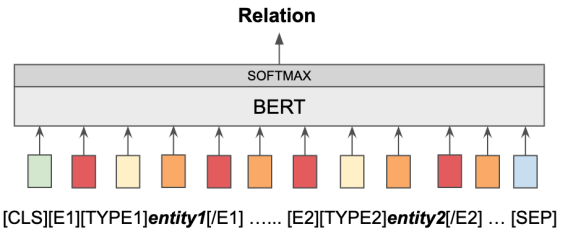


Figure 3.4: Entity and Type Markers

model input.

3.2.2 Models

This subsection presents different input configurations for the RE model. Following the work from Soares et al.[183], we will take BERT[44] pre-trained model and adapt it to solve our RE task. On top of BERT we add a Softmax classifier, which will predict the relation type (r). As baseline for comparison we use Soares et al.[183] configuration of BERT with *Entity Markers*. We will start by briefly overviewing their method, and continue with our two configurations proposed to add *Type Markers*.

3.2.2.1 Entity Markers (Baseline)

As stated in 3.1.1, candidate facts $(\mathbf{x}, \mathbf{e}_1, \mathbf{e}_2)$ contain a sequence of tokens from a sentence $\mathbf{x}$ and the entities span $\mathbf{e}_1$ and $\mathbf{e}_2$. *Entity Markers* (EM) are used to identify this entity span in the sentence. They are four special tokens $[E1_{start}]$, $[E1_{end}]$, $[E2_{start}]$ and $[E2_{end}]$ that are placed at the beginning and end of each of the entities, i.e.:

$$\hat{\mathbf{x}} = [x_0 \dots [E1_{start}] x_i \dots x_j [E1_{end}] \dots [E2_{start}] x_k \dots x_l [E2_{end}] \dots x_n]$$

this token sequence ($\hat{\mathbf{x}}$) is fed into BERT instead of $\mathbf{x}$. Figure 3.2 displays the described input configuration.

3.2.2.2 Type Markers only

A first solution to introduce *Type Markers* (TM) into the system is replacing the whole entity mention with the *Type Marker*. In this new configuration, there is no need to indicate the entity span. However, we still need to indicate which entity is performing as subject or object, because relations are directed. Thus, an *Entity Marker* for each entity is still needed: $[E1]$, $[E2]$. Figure 3.3 displays the model configuration, we use $[Type_{e_m}]$ to refer to each entity *Type Marker*. The modified $\mathbf{x}$ which will be fed into BERT looks

like:

$$\hat{\mathbf{x}} = [x_0 \dots [E1][Type_{e_1}] \dots [E2][Type_{e_2}] \dots x_n]$$

3.2.2.3 Entity and Type Markers

Finally we propose a combination of both previous models. It consists on adding *Type Marker* tokens without removing entity mentions nor any *Entity Marker*. Removing the entity mention, as in the TM only proposal, the model looses contextual information that may be useful for RE. In this configuration we try to provide the maximum information to help the prediction. The resulting input $\hat{\mathbf{x}}$, displayed in Figure 3.4, is:

$$\hat{\mathbf{x}} = [x_0 \dots [E1_{start}][Type_{e_1}]x_i \dots x_j[E1_{end}] \dots [E2_{start}][Type_{e_2}]x_k \dots x_l[E2_{end}] \dots x_n]$$

This model keeps the whole contextual information from the entity mentions, while adding the semantic types of the entities.

3.3 Experiments

The presented contributions for RE have been tested in an experimental set up. The different variations of the RE model, presented in Section 3.2.2 have been compared considering two datasets: the well known TACRED [219] dataset, and the new TypeRE[1] dataset [61] released during this dissertation.

3.3.1 Datasets

TACRED [219] is used with the purpose of comparing our system with other works. This dataset provides entity spans and relation category annotations for 106k sentences. Moreover, entity-types annotations for the subject and object entities are included. There are 41 different relation categories, plus the no-relation label, and 17 entity-types.

However, TACRED is not aligned to VKG, limiting the applicability of the trained model to cbe integrated into Vilynx's systems. To overcome this limitation, in this work we present the TypeRE dataset. This dataset is aligned with our ontology to be able to integrate the RE model into our knowledge graph population system. As manually annotating a whole corpus is an expensive task, we generated the new dataset by aligning three public RE datasets with our ontology. The datasets used are: Wiki80 [82], KBP37 [215] and KnowledgeNet[2] [129]. The entities from all three datasets were disambiguated to Freebase [25] identifiers. For Wiki80 and KnowledgeNet datasets, Wikidata identifiers are already provided, so the linking was solved by simply mapping identifiers. For KBP37 we disambiguated the annotated entities to Freebase ids using Vilynx's NERD system [59], as no identifiers are provided. For the three datasets, when an entity could not be disambiguated or mapped to a Freebase identifier, the whole sentence was discarded. For each entity, its root type is also added into the dataset. The included types are: "*Person*", "*Location*", "*Organization*", "*Work*", "*Occupation*" and "*Sport*". Sentences with entities with not known types were discarded. Regarding relations, we manually

[1] https://figshare.com/articles/dataset/TypeRE_Dataset/12850154

[2] Only training data annotations are publicly available

Table 3.1: Comparison of RE datasets. For each dataset we display the total number of sentences (Total), the number of sentences in each partition (Train, Dev and Test), the number of relational categories, and the number of unique entities labeled.

Dataset	#Total	#Train	#Dev	#Test	#Relations	#Entities
TypeRE	30.923	24.729	3.095	3.099	27	29.730
KnowledgeNet[129]	13.000	10.895	2.105	-	15	3.912
Wiki80[82]	56.000	50.400	5.600	-	80	72.954
KBP37[215]	20.832	15.765	3.364	1.703	37	-

Table 3.2: Test performance on the TACRED relation extraction benchmark.

	Dev			Test		
	Precision	**Recall**	**F1**	**Precision**	**Recall**	**F1**
ERNIE[221]	-	-	-	69.9	66.1	67.9
SpanBERT [100]	-	-	-	-	-	68.1
BERT$_{\text{EM}}$ [183]	65.8	68.4	67.1	67.8	65.3	65.5
BERT$_{\text{TM}}$	66.3	**71.0**	68.6	67.8	**69.4**	68.5
BERT$_{\text{EM+TM}}$	**69.6**	69.0	**69.3**	**70.3**	67.3	**68.8**

aligned relational categories from the datasets to our ontology relations. In order to make sure external dataset relations are correctly matched to ours, we validated that all triples in the dataset had a valid root domain and range given the relation, and discarded the sentences otherwise. Sentences from relations not matching our ontology and from relations with less than 100 annotated sentences, were discarded.

The dataset metrics are presented in Table 3.1. In comparison with the origin datasets. Type-RE is composed by 30.923 sentences expressing 27 different relations, plus the no-relation label, being a 73.73% of the total data from Wiki80, 19.85% from KBP37 and 6.42% from KnowledgeNet. The partition between train, develop and test sets was made in order to preserve an 80-10-10% split for each category.

3.3.2 Results

In this section we compare the proposed input configurations to combine *Type Markers* (TM) and *Entity Markers* (EM), against the baseline model, BERT$_{\text{EM}}$[183]. For all variants, we performed fine-tuning from BERT$_{\text{BASE}}$ model. Fine-tuning was configured with the next hyper-parameters: 10 epochs, a learning rate of 3e-5 with Adam, and a batch size of 64.

Table 3.2 presents the performance on the TACRED dataset. Our configuration combining *Entity* and *Type Markers*, BERT$_{\text{EM+TM}}$, exceeds the baseline (BERT$_{\text{EM}}$) by a 3.3% F1 and BERT$_{\text{TM}}$ exceeds it by a 3% F1, on the test set. The two proposed implementations also obtain better F1 score than ERNIE [221] and SpanBERT[100], when trained with base model. Some works [183, 100] have reported higher F1 scores with a larger BERT$_{\text{LARGE}}$ language model. The very high computational requirements of this model prevented us from providing results with them. However, published results [183] on our baseline configuration (BERT$_{\text{EM}}$) show promising possibilities to beat state-of-the-art when training our proposed models on BERT$_{\text{LARGE}}$.

Table 3.3 shows performance for the three input configurations on the TypeRE dataset.

Table 3.3: Test performance on the TypeRE relation extraction benchmark.

	Dev				Test			
	Precision	**Recall**	**F1**	**Accuracy**	**Precision**	**Recall**	**F1**	**Accuracy**
BERT$_{\text{EM}}$ [183]	84.3	86.9	85.6	90.9	87.0	88.3	87.6	92.1
BERT$_{\text{TM}}$	80.4	86.6	83.4	89.1	81.5	88.5	84.8	89.7
BERT$_{\text{EM+TM}}$	**88.4**	**87.0**	**87.7**	**93.2**	**90.2**	**89.5**	**89.9**	**93.7**

Our proposed configuration, BERT$_{\text{EM+TM}}$, achieves the best scores of the three configurations with a 2.2% F1 improvement over the baseline. However, BERT$_{\text{TM}}$ decreases overall performance in comparison to the baseline, while for the TACRED dataset it performed better. We believe this difference is because the granularity on the types given in TACRED (17 types) is higher than in TypeRE (6 types). This increased detail on types taxonomy provides a better representation an thus improved classification.

Regarding individual relations evaluation, we observed type information helps improving detection of relations with less training samples, as it helps generalization: e.g. "*PER:StateOrProvinceOfDeath*" and "*ORG:numberOfEmployees*", some of the relations with less data samples in the TACRED dataset, improve the F1-score by a 32% and 13% correspondingly when using BERT$_{\text{EM+TM}}$.

3.4 Conclusions

This chapter has presented opportunities for enhancing the quality of a RE system by combining IE techniques with Semantics. The relation extractor model proposed improves performance with respect to the baseline, by adding entity-types knowledge. To introduce types information, we have presented *Type Markers* and proposed two novel input configurations to add these markers when fine-tuning BERT. The proposed models have been tested with the widely known RE benchmark, TACRED, and the new TypeRE dataset, presented and released in this work. For both datasets, our models outperform the baseline and show strong performance in comparison to other state-of-the-art models.

As future work, we plan to study novel RE architectures which integrate knowledge graph information into the language model representation, inspired by [149].

Table 3.4: TACRED relations results

		BERT$_{EM}$			**BERT$_{TM}$**			**BERT$_{EM+TM}$**		
	Samples	**P**	**R**	**F1**	**P**	**R**	**F1**	**P**	**R**	**F1**
per:age	200	0.81	0.97	0.88	0.85	0.85	0.85	0.82	0.96	0.88
per:other_family	60	0.42	0.55	0.47	0.7	0.43	0.54	0.54	0.55	0.55
per:city_of_death	28	0.57	0.29	0.38	1	0.21	0.35	0.63	0.43	0.51
org:founded	37	0.71	0.95	0.81	0.78	0.86	0.82	0.73	0.86	0.79
org:top_members/employees	346	0.68	0.68	0.68	0.74	0.84	0.79	0.73	0.82	0.78
per:cause_of_death	52	0.6	0.17	0.27	0.85	0.33	0.47	0.62	0.31	0.41
per:parents	88	0.7	0.65	0.67	0.7	0.7	0.7	0.75	0.68	0.71
org:member_of	18	0	0	0	0	0	0	0	0	0
org:shareholders	13	0	0	0	0.67	0.15	0.25	0	0	0
org:political/religious_affiliation	10	0.43	0.3	0.35	0.67	0.2	0.31	0.62	0.5	0.56
per:religion	47	0.55	0.64	0.59	0.49	0.6	0.54	0.81	0.36	0.5
per:city_of_birth	5	0.25	0.4	0.31	0.4	0.4	0.4	0.67	0.4	0.5
per:stateorprovince_of_birth	8	0.43	0.38	0.4	0.38	0.38	0.38	1	0.38	0.55
org:founded_by	68	0.62	0.71	0.66	0.63	0.75	0.68	0.64	0.65	0.64
org:members	31	0.5	0.1	0.16	0.67	0.06	0.12	0.75	0.19	0.31
per:children	37	0.37	0.62	0.46	0.67	0.22	0.33	0.4	0.65	0.49
per:alternate_names	11	0.18	0.36	0.24	0.25	0.18	0.21	0.18	0.36	0.24
no_relation	12184	0.93	0.91	0.92	0.93	0.92	0.93	0.93	0.93	0.93
per:employee_of	264	0.51	0.5	0.5	0.69	0.78	0.74	0.73	0.7	0.72
org:subsidiaries	44	0.42	0.39	0.4	0.58	0.34	0.43	0.58	0.5	0.54
org:alternate_names	213	0.75	0.79	0.77	0.74	0.74	0.74	0.65	0.85	0.74
org:stateorprovince_of_headquarters	51	0.59	0.82	0.69	0.62	0.78	0.7	0.76	0.61	0.67
per:date_of_birth	9	0.7	0.78	0.74	0.86	0.67	0.75	1	0.78	0.88
org:country_of_headquarters	108	0.5	0.6	0.55	0.6	0.46	0.52	0.46	0.73	0.57
per:spouse	66	0.36	0.88	0.51	0.62	0.76	0.68	0.65	0.73	0.69
per:charges	103	0.61	0.87	0.72	0.71	0.72	0.71	0.82	0.8	0.81
per:cities_of_residence	189	0.41	0.74	0.53	0.56	0.69	0.62	0.58	0.51	0.54
org:dissolved	2	0.17	0.5	0.25	1	0.5	0.67	0.2	0.5	0.29
per:date_of_death	54	0.71	0.5	0.59	0.62	0.54	0.57	0.61	0.57	0.59
per:country_of_birth	5	0.25	0.4	0.31	0.2	0.2	0.2	0.33	0.2	0.25
org:parents	62	0.41	0.29	0.34	0.8	0.06	0.12	0.51	0.31	0.38
per:stateorprovinces_of_residence	81	0.41	0.65	0.51	0.57	0.72	0.64	0.59	0.54	0.56
per:title	500	0.86	0.77	0.81	0.77	0.9	0.83	0.84	0.85	0.85
per:schools_attended	30	0.75	0.6	0.67	0.61	0.73	0.67	0.82	0.6	0.69
per:countries_of_residence	148	0.51	0.28	0.37	0.5	0.42	0.46	0.5	0.37	0.43
org:city_of_headquarters	82	0.64	0.78	0.7	0.71	0.82	0.76	0.69	0.79	0.74
org:number_of_employees/members	19	0.68	0.68	0.68	0.73	0.42	0.53	0.67	0.63	0.65
per:origin	132	0.64	0.57	0.6	0.48	0.8	0.6	0.55	0.64	0.59
org:website	26	0.61	0.85	0.71	0.68	0.88	0.77	0.53	0.88	0.67
per:siblings	55	0.58	0.76	0.66	0.66	0.69	0.67	0.84	0.65	0.73
per:country_of_death	9	0	0	0	0	0	0	0	0	0
per:stateorprovince_of_death	14	0.33	0.14	0.2	0.67	0.14	0.24	0.71	0.36	0.48
micro average		0.61	**0.65**	**0.63**	0.68	0.69	0.69	0.68	**0.69**	**0.69**
macro average		0.5	0.54	0.5	0.63	0.52	0.53	0.61	0.55	0.56

Part II

Event Knowledge Graph

Introduction

4

An increasing amount of news documents are published daily on the Web to cover important world events. News aggregators like *Google News*[1] or *Yahoo! News*[2] help users navigate by grouping this overwhelming amount of materials in event clusters. Such systems facilitate users to stay informed on current events and allow them to follow a news story as it evolves over time. This aggregation task falls on the field of Topic Detection and Tracking (TDT), which aims to develop technologies that organize and structure news materials from a variety of broadcast news media. However, media professionals are in need of more advanced tools to describe, navigate and search specific pieces of information before writing their own piece of news. Semantic Web and Information Extraction (IE) technologies provide high level structured representations of information, which can help solving the mentioned problems.

In this second part of the dissertation, we describe VLX-Stories, a system under exploitation that alleviates the aforementioned issues from journalists teams. It consists of a unified online workflow of event detection (Chapter 5) and representation (Chapter 6), with the aim of building an event-based knowledge graph. In VLX-Stories, events are represented by means of an ontology inspired on the journalist Ws [182] (*what* is happening, *who* is involved, *where* and *when* it took place) plus the general *topic* under discussion. The system is characterized by the adoption of semantic technologies, combined with IE techniques for event encoding. The extraction of mentions and its linkage to entities from an external multilingual knowledge graph generates an event linked space. This allows the multilingual linkage across stories, semantic search, and the linkage to external contents by matching entities.

In this first chapter we introduce related work (4.1) on event-encoding systems, we follow with an overview on the whole VLX-Stories System (4.2) and finish with a summary on this part contributions (4.3).

4.1 Related Work

The system presented in this work tries to solve the *semantic gap* between the coverage of structured and unstructured data available on the Web [150], in order to provide journalistic tools for event analyzing. In the past decades, a great amount of research efforts has been devoted to text understanding and Information Extraction (IE). Many research projects have entangled with the different problems described in this work, i.e. news aggregation [83, 114, 37, 187, 75], event pattern extraction [218, 95], event ontology population [163, 205] and automatically answering journalist Ws [79, 78]. However, only

[1] http://news.google.com
[2] http://news.yahoo.com

a few big projects are comparable to our system as end-to-end online pipelines for event detection and encoding. In this section we will focus on reviewing these large-scale systems.

Two well-known event-encoding systems are the *Integrated Crisis Early Warning Systems*[3] (ICEWS) and the *Global Database of Events, Language and Tone*[4] (GDELT). This two projects have been developed to automatically extract international political incidents such as protests, assaults and mass violence from news media. These datasets are updated online, making them useful for real-time conflict analysis. ICEWS is a project supported by the Defense Advanced Research Projects Agency (DARPA), to be used by US analysts. Its data has recently been made public though Harvard's Dataverse[5], however events are posted with a 1 year delay and the techniques and code utilized are not open source. GDELT was built as a public and more transparent version of ICEWS. Its data is freely available and includes over 200 million events since 1979, with daily updates. However, legal controversies over how data resources were obtained distanced it from research. It is currently incorporated into Google's services and its data is utilized for analysis of international events [115, 110]. Since ICEWS and GDELT are the two most widespread news databases, several comparison studies have been made between them. Even though no conclusion could be extracted on the superiority of any system, GDELT overstates the number of events by a substantial margin, but ICEWS misses some events as well [201, 198].

A more recent event data program is the *Open Event Data Alliance*[6] (OEDA). This organization provides public multi-sourced political event datasets, which are weekly updated [174]. All the data is transparent and they provide open code of the ontologies supported. They use Standford CoreNLP tools [126] and WordNet[130] dictionaries. However, OEDA's efforts still have not reached the scale of the other two mentioned projects.

Another well-known project is the *NewsReader*[7][196]. This system is a big collaborative research project, which constructs an Event-Centric Knowledge Base (ECKB) based on financial and economic news articles. They take advantage of several public knowledge resources to provide multilingual understanding and use DBpedia [15] as knowledge graph for EL. They define their own event ontology, the Simple Event Model (SEM) [194], which is designed to be versatile in different event domains allowing cross-source interoperability. To deal with entities not properly represented in the knowledge resources, they introduce the concept of *dark entities*.

From the works presented, ICEWS, GDELT and OEDA are focused on political data for the analysis of conflicts, and NewsReader generates an ECKB from financial data. Notice there is still a big coverage gap when it comes to media event encoding. In this sense, VLX-Stories offers a wider service for journalistic purposes, as it covers, as well as politics and finances, many other categories, like sports, entertainment, lifestyle, science and technology.

[3] https://www.icews.com/
[4] https://www.gdeltproject.org/
[5] https://dataverse.harvard.edu/dataverse/icews
[6] http://openeventdata.org/
[7] http://www.newsreader-project.eu/

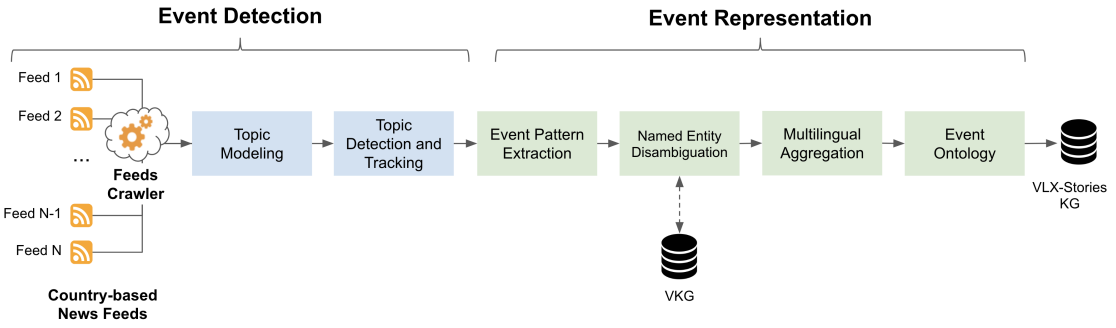


Figure 4.1: Pipeline schema of VLX-Stories framework.

4.2 System Overview

VLX-Stories system, proposed in this part of the thesis, has been designed to provide a solution to some of the main challenges of Topic Detection and Tracking (TDT). The features described bellow make our system different from any system found on literature:

- Work on real time: many works found on literature solve TDT on static news corpus where topics are detected for the whole year or daily, e.g. [128, 152] are doing daily story clustering. Nevertheless, we require from our system to update event clusters on real time in order to track updates on the news. For that purpose we will design a crawler of news feeds that will provide constant updates on new documents being published.
- Not topic restricted: several works and databases which are restricted to collection of topics, like [165, 189]. However, as our system is designed for journalistic purposes, it needs to be open to any topic.
- Generate multilingual event clusters: we want to track event development across countries. For that, multilingual event clusters will be constructed on top of the monolingual ones by using its semantic event representations for linking. This approach is similar to [83, 152, 165].

Following, we provide a general overview on VLX-Stories pipeline. As can be seen in Figure 4.1, it is split in two major blocks that we briefly summarize in this section. The underlying architecture of such blocks that allows us to accomplish the listed goals, will be further described in the following chapters.

The first block of the pipeline (**Event Detection**) extracts news articles from media feeds and aggregates them into clusters describing the same event. Articles are collected by an RSS feeds crawler, and represented as vectors in the Topic Modeling module. Then, these articles are associated to an already detected event (Topic Tracking) or used as a seed for a new event (Topic Detection). The output of this block are clusters of aggregated news articles representing distinct world events. However, there is not semantic understanding or representation, making them difficult to search and link.

The second block (**Event Representation**) encodes the events by synthesizing the agents, locations and actions involved in each cluster. This is achieved by extracting

the entities involved in each event and structuring the knowledge in an ontology which answers the journalist W's [182] and the Topic. To do that, a keyword-based pattern is extracted for each detected event. Then, in the Named Entity Disambiguation module, mentions from the pattern are mapped to entities from the VKG, introduced in Section 0.7. As knowledge graph entities are language agnostic, the entities from each event semantic pattern can be used to map language-based event clusters and aggregate them into multilingual ones. Finally, entities are structured into the Event Ontology using their respective types and relevancy.

4.3 Contributions

The final system deployed on production aggregates news articles from over 4,000 RSS feeds, processing an average of 17,296 articles/day and detecting over 350 worldwide events/day from seven different countries (United States, Spain, Canada, Australia, Ireland, United Kingdom and Portugal) and three languages (English, Spanish and Portuguese).

The contributions of this part of the dissertation can be summarized as: a) the construction of an online and multilingual news aggregation system, b) the representation of events and the construction of an event knowledge graph and c) the large-scale deployment of the system, which is currently consumed by several media companies to gather information more efficiently.

Event Detection

5

The rise of news aggregator sites is a notable phenomenon in the contemporary media landscape. Many audiences have started consuming news from such aggregators, such as *Yahoo News*, *Google News*, and the *Huffington Post*, instead of traditional media sources.

The construction of such systems falls in the field of Topic Detection and Tracking (TDT). Such research field initial motivation was the monitoring of broadcast news and alert analysts to new and interesting event happening in the word [10]. It provides tools to deal with the overwhelming volumes of information being daily published by detecting what it is called *topics*. Within TDT, a topic is defined as a set of news stories that are strongly related by some seminal real-world event, i.e. when a bomb explodes in a building, that is the event that triggers the topic. Any news stories that discuss the explosion, rescue attempts, the search of perpetrators, arrest, etc. are all part of the topic.

In this chapter we present the first module of VLX-Stories system. It consists on an online news aggregator, which retrieves multi-regional and multi-lingual articles from news sites, and aggregates them by topic. Among the different TDT tasks, we focus on: *cluster detection* and *tracking*. Cluster detection is the problem of grouping stories as they arrive, based on the topics discussed, while tracking refers to monitoring the stream of news stories to find additional stories on a topic that has been already identified.

First part of this chapter introduces background knowledge and related work on TDT (5.1). We follow with an overview on our news aggregator methodology (5.2), and experimental evaluation on the clustering quality (5.3) and we provide analytics on the system performance (5.4). Finally we present some conclusions (5.5) on the system developed.

5.1 Background

This section reviews research on the TDT field. From this field, we will deal with the next three research lines: topic modeling (5.1.1), topic detection or clustering (5.1.2), and topic tracking (5.1.3). In the next sub-sections these subtasks are described with examples of how different state-of-the-art models solved it.

5.1.1 Topic Modeling

The task of topic modeling consists in representing the abstract topic that occurs in a collection of documents in a way that allows to compute mathematical operations like distance or similarity. Topic modeling for text representation have been largely studied by the NLP community. Several methods and models of document representation have

been proposed for TDT, i.e. word-based [169], language models [151], and graph-based [36] representations. Classical topic modeling for TDT relays in bag-of-words (BoW) alike representation, like TF-IDF (term frequency - inverse document frequency), which weights words relevancies based on its appearance frequency on the whole corpus and in the document. Another classical method is the use of Latent Dirichlet allocation (LDA) [24], in this model each document may be viewed as a mixture of various topics where each document is considered to have a set of topics that are assigned to it via LDA. Probabilistic latent semantic analysis (PLSA) [90] and Laplacian PLSA (LapPLSA) have also become some of the most commonly used topic modeling techniques. However, news stories typically revolve around people, places and other name entities. Shah [176] showed that using these named entities leads to better TDT performance than using all words. Thus, adaptations of TF-IDF into weighted bag-of-concepts have proved to give higher performance for TDT task. e.g. SPIGA [83] represents documents by doing EDL and constructing a weighted bag-of-concepts with these entities. Also, graph based models like and-or-graphs (AOG) [120] or graph regularization methods [32] have recently been proving high accuracy results and are useful for multimodal topic modeling. For example, in [120] a novel representation using a Multimodal Topic And-Or Graph (MT-AOG) is presented.

5.1.2 Topic Detection

Topic Detection consists on generating clusters of distinct documents reporting the same story. This task is characterized by the lack of knowledge of the event to be detected and of the amount of clusters to divide the space in. Therefore, methods that require to define the number of cluster a priori, like k-means, are not useful for this task. However, other traditional document clustering methods based on density or hierarchies, like DBSCAN [52] or hierarchical agglomerative clustering [83, 3, 186] have proved great performance. Usually this task is unsupervised, but semi-supervised approaches have also been proposed for this task, e.g. in [37], event clustering is performed by using Reuter's articles as clustering 'seeds to generate top-level clusters and uses agglomerative clustering to gather documents into distinct sets.

5.1.3 Topic Tracking

The traditional topic tracking problem is defined as the task of associating incoming documents with events known to the system. An event is defined by a set of associated documents that discuss the same story. Thus, each target event is defined by a list of documents that discuss it. Many methods have been proposed for solving this task. The most straight forward and used method is to use a k-NN to assign new documents to already existing story clusters based on distance metrics. However, deciding the topic for each incoming story based on the previous learned topics can take a long time in a large data collection. Some models incorporate time information, such as Dynamics Topic Model [23], and the temporal Dirichlet mixture model (TDPM) [8]. e.g. in SPIGA [83] cosine-similarity is used to measure distance between new documents entering and the centroid vector of existing topics. They also add a time factor which favors newest documents to be assigned to more recent event. If a document's similarity to all clusters is lower than a predefined threshold, a new cluster is created.

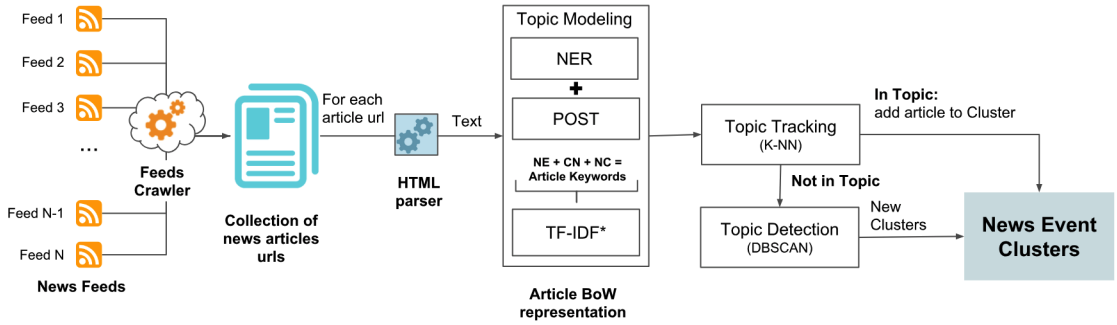


Figure 5.1: Schema of the news aggregator system.

5.2 Method

This section describes the three parts of the news aggregation system, outlined in Figure 5.1. First, a collection of news articles is built by crawling RSS feeds and parsed to extract all its text data (5.2.1). Afterwards, in the topic modeling block (5.2.2), articles are represented with its entities following a bag-of-words (BoW) alike approach. Finally, in the topic tracking module, each article vector is compared with articles grouped in event clusters: if matching the event, it is assigned to the cluster; if not, it is added to a pool of articles that will be processed in the topic detection module (5.2.3) in order to detect newly emerging events.

5.2.1 News Feeds Crawler

News articles are collected by an RSS feeds crawler, which processes 1500 news feeds every 30 minutes. The RSS feeds come from a manually generated list of 4162 feeds from the main media sources of seven countries: United States, Australia, Spain, Canada, United Kingdom, Portugal and Ireland. Feeds are also manually categorized in seven category groups: *politics*, *sports*, *general news*, *lifestyle and hobbies*, *science and technology*, *business and finance*, and *entertainment*. The feeds crawler visits each feed, crawls it, and stores in the DB each article URL, publication date, title and description if provided. In a second step, whenever a new article is detected in a feed, we crawl the URL and parse the article using a customized HTML parser to extract all its text data and images.

5.2.2 Topic Modeling

Topic modeling consists of representing the abstract matter that occurs in a collection of documents. To do this, we will rely on a BoW representation of the articles. As news stories typically revolve around people, places and other named entities (NE), some works [83, 176] use mentions of NE instead of all words in the document. However, some news do not turn around NE, e.g. weather news or events related to anonymous people. Therefore, other information, such as common nouns (CN) or noun chunks (NC), is needed to distinguish these kinds of events [140]. Combining these three extractions, we will use named entities, common nouns and noun chunks in the BoW representation, instead of all words in the text corpus. We will call this collection of mentions and nouns *article keywords*. These keywords are extracted from the article's text by a Named Entity Recognition (NER) module and Part of Speech Tagger (POST), introduced in previous

part of this dissertation (2.1). For performance reasons, we constraint the articles to be represented for at least 8 keywords, and a maximum of 80 keywords.

BoW keyword's frequencies are weighted by a slightly modified TF-IDF (term frequency - inverse document frequency), which reflects how important a word is to a document in a collection or corpus. TF-IDF is computed as the product of the term frequency (f_k) by the inverse document frequency (idf_k). However, we tune the TF with a weight to give more relevance to those keywords appearing on the title (α), description (γ), or that are NEs (β). Finally, inspired by [83], we apply a time factor with a linear function, which favors news documents to be assigned to more recent events.

5.2.3 Topic Detection and Tracking

Once a new article is ingested by the system, we must detect if it is associated to an already detected event (topic tracking) or it describes a new event (topic detection). The system follows the next two steps algorithm:

1. First, we try to associate the incoming new article a_i to already created article clusters. For this topic tracking task, we use the *k-Nearest Neighbours* (k-NN) algorithm. Thus, we associate the article with an event cluster if there are more than k articles in the cluster with a similarity higher than a given threshold ϵ. We use *cosine similarity* as distance metric for the k-NN algorithm.

2. If the incoming article a_i is not associated to any event cluster, we try to build a new cluster with other articles not yet related to any event. This is the task of topic detection. The chosen clustering technique for topic detection is DBSCAN [52], which is a unsupervised density-based algorithm that provides robustness against the presence of noise. This method requires the estimation of two hyper-parameters: *min samples*, which is the minimum number of samples needed to generate a new cluster, and *eps*, the maximum distance allowed within its samples. We decided to fix the $minsamples = 5$, thus all events are represented with at least five articles, and we optimize *eps* in order to have high precision without missing many events. As well as topic tracking, we use use *cosine similarity* as the distance metric.

Moreover, some design decisions were made in order to compensate some of the problems of dealing with an online and large-scale deployment application with noisy Web data. In order to prevent wrong event detections due to web parser errors, we added two extra conditions on the cluster generation:

- The clustered articles need to be from at least three different news publishers, and one media publisher can not own over 50% of the articles in a cluster. This condition aims to detect relevant events (several publishers talk about it) and to prevent the detection of events biased towards the opinion of a single news publisher. Moreover this diversity helps preventing the creation of events based on fake news. Values were chosen after manually analyzing several detection errors.

- Also, speed issues had to be considered to provide real-time tracking on news events, as the amount of comparisons between articles grows quadratically with the number of articles, slowing the whole article comparison. We decided to cluster

Table 5.1: Results of the news event detection on UCI Dataset subset

	#articles	#events	Event P.	Article P.	Article R.	Article F1
Business	21,535	1,796	78.28	91.44	51.76	66.11
Entertainment	36,790	1,673	88.76	96.43	64.44	77.26
Technology	23,921	1,811	85.69	94.38	57.37	71.36
Health	9,482	1,009	68.18	94.86	56.85	71.09
GLOBAL	**91,72**	**6,28**	**81.58**	**94.57**	**58.54**	**72.32**

articles by country, and for those countries with more feeds, we use a category-based comparison between articles. The category of the feed is used for this split, and in case the feed provides general news from any category, we trained a deep classifier based on a one layer LSTM [85] to predict the article category from its title. The training dataset was constructed by merging the category titles from the UCI-ML News Aggregator Dataset [45] and titles from the manually labeled RSS news feeds.

5.3 Experiments

Regarding news aggregator or event detection evaluation, we used a subset of the UCI-ML News Aggregator Dataset [45]. This dataset consists of 7,231 news stories, constructed from the aggregation of news web pages collected from March 10 to August 10 of 2014. The resources are grouped into clusters that represent pages discussing the same news story or event. Events are categorized into four different categories: entertainment, health, business and technology. For each news article, its title, URL, news story id and category are given. However, we had to discard 13% of the events on the dataset because of the following reasons: 36% of the URL link's were broken, our system could not extract enough keywords from 17% of the articles, and some of the remaining news stories were not represented by enough articles to be detected by our system (each of our events needs to be represented by at least 5 news articles).

The final dataset subset consists on 6,289 events constructed by 91,728 news articles. For our experiments the DBSCAN parameters were set to $eps = 0.65$ and $minsamples = 5$.

Table 5.1 presents the news event detection results on the dataset. Most of the events are correctly detected (81.58%), however a lot of articles are not associated to any event. This is reflected by the high average precision (94.57%) but a poor recall (58.548%), which translate to an article classification F1 score of 72.32%. This is mostly because of the restrictive parameters set in the system in order to make sure that aggreagated news served are correct. Quality of aggregated news is similar across news categories. The lowest quality is found in the business category, because most of the business news share financial terms which are repeated in many articles, even not related ones. Best results are for entertainment, the type of news with more named entities, which improves the representation.

5.4 Analytical Results

VLX-Stories was first deployed on July 2018 for its use in the United States. Since then, the system has been growing, adding new world regions and languages. Table 5.2

contains the activation date of VLX-Stories for each country, the language it processes, the number of feeds activated on each country, the average number of events detected each day and the daily number of articles processed and associated to events. Results are provided country by country and also on the worldwide event aggregation. According to these statistics, VLX-Stories grows in a speed average above 300 news events/day, classifying an average of over 17k multilingual articles from seven different countries. Since we activated the multi-regional event aggregation module on November 2018, the system includes the option of analyzing how an event is reported in different world regions and in different languages. Semantic disambiguation is essential for this multilingual aggregation task.

Table 5.2: Statistics on VLX-Stories Event Detection module.

Country	**Activation date**	**Language**	**#Feeds**	**Events/Day**	**Articles/day**
USA	07/2018	English	952	96.70	4,745.59
SP	08/2018	Spanish	918	90.61	4,929.01
CA	09/2018	English	551	27.15	990.04
AU	09/2018	English	893	53.19	3,223.09
IR	09/2018	English	121	20.46	654.25
UK	09/2018	English	442	38.57	1,518.63
PT	09/2018	Portuguese	285	35.80	1,235.76
TOTAL	-	-	**4,162**	**362.48**	**17,296.37**
World Wide	11/2018	Multilingual	4,162	301.84	17,296.37

5.5 Conclusions

This chapter has presented an online event detection system which aggregates news articles from over 4,000 RSS feeds. The engine presented combines topic modeling and TDT techniques to represent news articles and cluster them into aggregated news stories. The system is deployed in production and processes an average of 17,296 articles/day, detecting over 350 worldwide events/day from seven different countries (United States, Spain, Canada, Australia, Ireland, United Kingdom and Portugal) and three languages (English, Spanish and Portuguese).

Our experimental results on the News Aggregator Dataset [45] show a correct detection of more than 80% of the events, and F-1 score of 72.32% for article classification. These numbers translate to a good event detection capability, but a poor recall when assigning articles into event clusters (58.54%). As a commercial tool, the system parameters have been optimized to prioritize precision over recall in order to serve correct information. However, future work should focus on improving article classification recall to provide a more complete event tracking.

Event Representation

6

The amount of event-centric information regarding global importance events, such as *Covid-19* or the *Ukranian War*, constantly grows on the Web within news sources and social media. Efficiently accessing and analyzing large-scale event-centric and temporal information is crucial for a large variety of real-world applications, e.g. Question Answering [89], timeline generation [11], cross-cultural studies [72], etc. The synthesis of event information into structured knowledge resources can help automatize all these applications.

News aggregators, like the one described in Chapter 5, provide large amounts of articles and event related contextual information in an automatic manner. Nevertheless, such systems lack on structured knowledge and require further processing to synthesize the information needed for the mentioned applications. The goal of this chapter is the extraction of such structured representation from aggregated news, in order to complete the VLX-Stories pipeline and create an event knowledge graph.

To this end, we analyze event representation techniques. These techniques try to synthesize the agents, locations and actions involved in an event in a formal machine understandable way, but still natural for humans. In this chapter we describe the design decisions and method for event representation and the qualitative experiments performed to evaluate it. Moreover, as mentioned on previous chapters, VLX-Stories is deployed in production, serving structured news stories on real time. We present the user interface designed for it.

This chapter is structured as follows. We start introducing background knowledge on event knowledge graphs and event schema (6.1). We continue by describing our method to represent events (6.2). Following, we present an evaluation experiment (6.3) and the user interface where such system is integrated (6.4), providing real time event-centric knowledge. We finish presenting conclusions (6.5) on VLX-Stories system.

6.1 Background

Knowledge graphs have gained increasing popularity in the past years thanks to their adoption in search engines like Google, Yahoo or Bing. This knowledge resources, usually store bibliographical facts about entities (e.g. person, organizations), like birth date and birth place. However, such information only represents a very small part of what happens in the world. Furthermore, these repositories tend to represent the actual state of the world and do not provide dynamic information and changes over time. Event information reported on daily news is rarely covered in such resources and tends to fade out. Nevertheless, temporal event-centric information can be of great importance for

professionals needing to reconstruct the past, analyze events and be aware of worldwide events.

In this section we will review related work on event-centric knowledge graphs (6.1.1) and event representation methods (6.1.2).

6.1.1 Event Knowledge Graphs

The lack of structured resources containing event-centric information, has recently motivated the research on the automatic extraction of events and the generation of event knowledge graphs. To construct such resources, two different perspectives are usually taken: aggregating event information from already structured knowledge resources (e.g. Wikipedia or DBpedia), and the creation of event knowledge graphs from unstructured information (e.g. news articles).

The first approach is the one taken by EventKG [71]. This system generates a multilingual event-centric temporal knowledge graph by mixing the event information found in several large-scale knowledge graphs and semi-structured sources. Such system incorporates over 690 thousand contemporary and historical events and over 2.3 million temporal relations. Nevertheless, this kind of methods provide static information and depend on external structured information. This approach is thus not applicable to create event resources where information is updated on real time.

The second kind of methods build event knowledge graphs directly from plain text news. These systems apply IE methods, like named entity disambiguation (NED) [83, 114, 196], to synthesise the agents and locations related to an event. Moreover, the use of entities to describe events is useful to solve co-reference and link multilingual articles. An example of method to construct an event knowledge graph of such kind is the ECKG [163]. In this work, they construct knowledge graphs in several distinct domains by extracting structured information (triples) from several news collections, using different NLP techniques like Named Entity Disambiguation (NED) and Semantic Role Labeling (SRL).

In this thesis we focus on the creation of an event knowledge resource, similar to the second approach. As we will see, we propose the use of IE techniques combined with semantic technologies to construct VLX-Stories KG.

6.1.2 Event Schema

Event representation intends to encode event-related information in a structured manner. It poses a multitude of challenges due to the variety of event domains, types, definitions and applications, e.g. summarization [153], triple representation [218, 119], pattern extraction [117, 21] or ontology population [163, 205]. However, in general, all these approaches revolve around extracting the main information units: "*what is happening*", "*who is involved*", "*where is it happening*" ,"*when did it happen*" and "*why did it happen*" . As mentioned before (Chapter 4), these information units are commonly known in journalism as the Five Ws [223] and are going to be the base of our event representation.

Most famous event ontologies or schemas try to synthesize these information units as

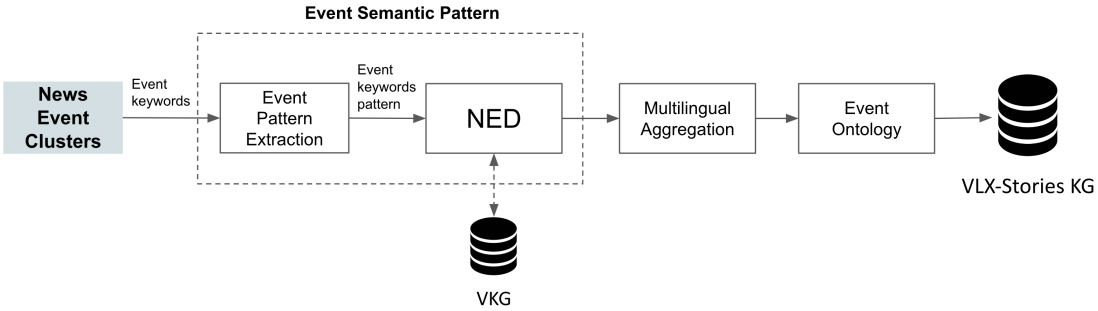


Figure 6.1: Pipeline schema of the event representation modules

well. For example, the Simple Event Model (SEM) [194] (used by ECKG), includes predicates like *hasActor* and *hasTime* to describe such information units. The ABC ontology [111] describes event-related concepts such as the situation, action, agent and their relationships. NOEM [199] introduces the additional journalistic component of "*how did it happen*" and adds it to the 5Ws model, calling it 5W1H.

6.2 Method

The event representation system has three differentiated parts, as can be seen in the diagram in Figure 6.1. The input of the system are clusters of aggregated news articles. The articles in each group are all in the same language and from the same region. For each one of the articles, we already have its representation as a bag of keywords, as previously described in sub-section 5.2.2.

The first part of the event representation framework focuses on the extraction of the most relevant entities describing an event (6.2.1), which we call the *event semantic pattern*. The secdond part consists in a module which, using such patterns, aggregates the regional-based events into worldwide events and ranks entities according to its relevance describing the event (6.2.2). Finally, the third module structures the extracted entities into a set of properties which summarize the main agents, locations and topics involved, by answering the four of the five journalistic Ws (*who*, *what*, *where*, *when*) and the main story *topic* (6.2.3). Notice that we are not answering the *why* because it requires a deeper understanding of the story and also there may not be a correct answer for it. Next subsections will describe the technical detail regarding each module.

6.2.1 Event Semantic Pattern

The extraction of the event semantic pattern is achieved thanks to the two modules depicted inside the Event Semantic Pattern block in Figure 6.1: *event pattern extraction* and Named Entity Disambiguation (NED). Frist, the event pattern extraction module finds the most relevant keywords describing the event (6.2.1.1), and the EL module disambiguates these keywords to entities from the VKG (6.2.1.2).

6.2.1.1 Pattern Mining

We first use a pattern mining approach to detect the most descriptive set of keywords for a given event. It is inspired by [218], where multimodal patterns are extracted to describe some predefined event categories with semantic concepts. Similar to this method, we search for the pattern of semantic concepts using a pattern mining approach which explodes the redundancy between aggregated news articles.

Data mining techniques search for patterns in data that are representative and discriminative. We define our pattern mining task with an *association rule* approach [7], such that our pattern corresponds to a set of association rules, $t^* \rightarrow y$, that optimize the *support* and *confidence* constraints for each event. Let n be the set of all keywords in the corpus $C = \{k_1, k_2, ..., k_n\}$; and a *transaction* A be the set of keywords from a given article, such that $A \subseteq C$. Given a set of m transactions belonging to the same event $T = \{A_1, A_2, ..., A_m\}$, we want to find the subset of C, say t^*, which can accurately predict the belonging to a target event $y \in E$. The *support* of t^* is an indicator of how often t^* appears in T, and it is defined as the proportion of transactions in the transaction set T that contain the itemset t^*:

$$s(t^*) = \frac{|\{A_a | t^* \subseteq A_a, A_a \in T\}|}{m} \tag{6.1}$$

Our goal is to find association rules that accurately predict the belonging to an event, given a set of keywords. Therefore, we want to find a pattern such that if t^* appears in a transaction, there is a high likelihood that y, which represents an event category, appears in that transaction as well. We define the *confidence* as the likelihood that if $t^* \subseteq A$ then $y \in A$, which can be expressed as:

$$c(t^* \rightarrow y) = \frac{s(t^* \cup y)}{s(t^*)} \tag{6.2}$$

Inspired by [117] we use the popular apriori algorithm [6] to find patterns within the transactions. We only keep the association rules with confidence $c_{min} \geq 0.8$ and calculate the support threshold (s_{min}) that ensures at least 10 keywords in the rule. Finally, we select the rule t^* with more keywords associated. These keywords are the ones that will be disambiguated into VKG entities.

6.2.1.2 Named Entity Disambiguation

The event keywords in the pattern will be mapped to entities in VKG. This is the second part of IE task called Entity Linking (EL), which consists on Named Entity Disambiguation (NED), described in Section 2.1.2. Entity disambiguation has been used in other event detection works, like [83, 114, 196], as it is useful to solve co-reference, link multilingual articles and it is a more sophisticated and standard knowledge representation.

In this work, the system used for disambiguation is the one presented in Section 2.2 of this dissertation. As described, the NED module gets entity candidates from VKG for each incoming mention. Entities are retrieved based on similarity matching between the text mention and the entities alias. Then, disambiguation is applied by scoring each candidate. Outputting this part of the system, we have a set of entities describing an event, which we call the *event pattern*.

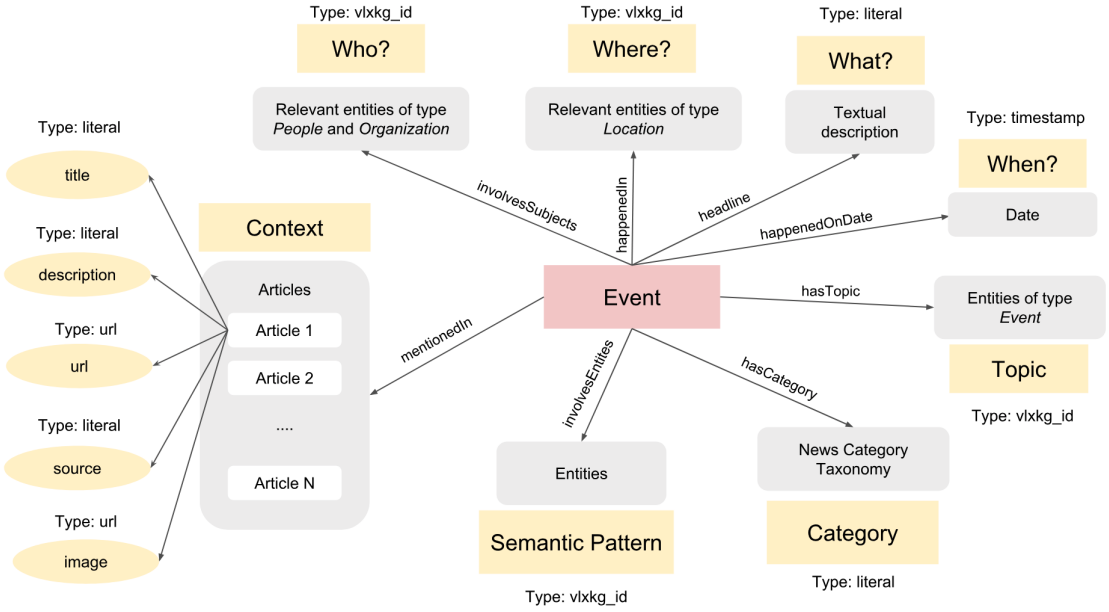


Figure 6.2: Event ontology schema.

6.2.2 Worldwide Event Detection (Multilingual Aggregation)

Before the final event modeling, we unify country-based aggregated news into worldwide-events. This consists on the *multilingual aggregation* module from the diagram in Figure 6.1. To do that, we will use the extracted *event patterns* of each news cluster to compare and merge events in case of match. We first rank the entities in the pattern by relevancy describing the event. The ranking is based on re-scoring entities based on its original keywords appearance frequency and origin (title, description or text body). As we solved the entity disambiguation, we recompute the entity frequency taking into account co-references. Origins are taken into account by weighting the frequency of appearance by the origin. Afterwards, country-events are represented with a bag of concepts BoC where entity relevancies are the weights. Cosine similarity is computed between country-events, and DBSCAN [52] is used to detect clusters of worldwide-events among the country-based events.

6.2.3 Event Ontology

Both semantic and contextual event information extracted on previous steps are processed in order to represent the collected data in an ontological manner. This ontological information is stored in VLX-Stories KG, which keeps growing with the multiregional news information provided by the feeds. In this section, we first motivate the modeling decisions we took designing the ontology and we continue by describing the information extraction process.

6.2.3.1 Modeling VLX-Event Ontology

The main requirement of our event ontology is that it has to synthesize in a both machine and human readable way the unstructured semantic pattern extracted. Events are usually defined by its agents, locations and actions occurring and the moment when the event

takes place. Journalistic articles are typically structured to answer four of the journalist 5W-questions: e.g. *what* happened, *who* is involved, *where* and *when* did it happen. These questions should be addressed within the first few sentences of an article to quickly inform readers. The fifth W, *Why* it happened, is often addressed in opinion columns or post-event coverage [101]. Moreover, news stories often fall into bigger topics which are composed by several events, like *Covid-19*, *Brexit*, *Academy Awards*, *Olympic Games*, etc. This information, if present, offers the possibility of tracking long story lines and to provide a complete context on the development of an event in a point in time.

Considering the above mentioned 4Ws and the topic, we defined our ontology with the next *properties* or *core classes* for each event: Who, What, When, Where and Topic. These properties will be extracted from the event semantic pattern and the titles and descriptions from the event articles. Moreover, as shown in the ontology schema of Figure 6.2, all entities in the semantic pattern will be included in the ontology within the Semantic Pattern class, answering or not the 4Ws or topic. The event Category, e.g. sports, politics, entertainment, etc.; is also included as a property. Additional context of the event is given by the clustered articles, from which we store the title, description, URL, news source and image, if present.

6.2.3.2 Event Properties Extraction

As the last step, all the information collected for each event is structured in the presented VLX-Event Ontology as followingly described:

- The Who, Where and Topic are extracted from the *event semantic pattern* using a set of filters based on entity types. Entities in VKG have a schema[1] type associated, which denotes if the entity is a person, place, organization, etc. These types will be used to classify the entities in the pattern together with the type of metadata field from which they were extracted. For this task only the entities from the title and description are used. Moreover, the same entity relevance scores computed for multi-regional event matching will be used to pick those most relevant entities. For each property we apply the next rules: the Who property needs to be an entity of type *Person*, *Organization* or *CreativeWork*; the Where is a property of type *Place* and the topic of type *Event*.
- We define the What of an event with a sentence (literal) summarizing the main event occurring. As news article's titles should give this information, we answer the What with the most descriptive title between the clustered articles. This is selected using a voting algorithm where each word in the title sums its given score in the semantic entities ranking. This approach favors longer titles, which also tend to be semantically richer.
- To answer When, we take the date of the first article published related to an event. We plan on improving it on next versions by analyzing time expressions in the text.
- Finally, we complete the ontology by adding the event Category. Categories come from our pre-defined set of categories, e.g. *Sport*, *Entertainment*, *Finances*, *Politics*, etc. The categories assigned to the RSS feeds are used to extract this information. One event may belong to more than one category.

[1] https://schema.org/

6.3 Experiments

To evaluate the semantic patterns, we measured the quality of the Named Entity Disambiguation (NED) of the keywords on the event pattern to semantic labels. The experiments were conducted over a corpus of 100 semantic event patterns randomly selected from the United States events, detected by our news aggregator module during the week from the 1st of January to the 7th of January 2019. The keywords from the patterns were mapped to entities from VKG using the NED module. The correctness of the mapping was evaluated with TP when the semantic entity related to the keyword was correct, FP when the semantic entity related was wrong, TN when no entity was related and it is not an existing entity or it is an error from NER, and FN if no entity was mapped but there is an entity in VKG for it. Results are displayed in Table 6.1, showing a total accuracy of 86%. However some mentions do not disambiguate to its correct entities. This is specially common when finding homonym words or unknown contexts. Further research should be developed to improve these ambiguous cases.

Table 6.1: Results on Entity Linking

#Event Patterns	TP	TN	FP	FN	Precision	Recall	F1	Accuracy
100	966	329	52	156	0.86	0.94	0.90	0.86

6.4 VLX-Stories User Interface

Media publishers such as news broadcasters, magazines, media companies and bloggers have the need to monitor world events in order to be aware of trends and events worldwide. News aggregators have provided a solution to navigate and consume news by grouping the overwhelming amount of articles published on event clusters. However, these tools are designed for general users and do not provide several crucial capabilities for media publishers: a long term view and context on the news stories, multi-regional and multi-lingual information or linkage to their contents or those of their competitors. Semantic Web and Linked Data technologies provide solutions which can be applied to the mentioned problems by using knowledge graph and ontologies to link, structure and serve this information.

In this section we present VLX-Stories's interface which is used in the editorial process. The presented user interface is used to access and query VLX-Stories, that encodes over 9000 events per month. This interface leverages semantic technologies to provide a complete linked space which allows navigation among time, categories, regions, publishers, topics, places or personalities. It is deployed in production and has been used by major media networks, accelerating the editorial process and improving their operational efficiency by helping on content discovery, search, content generation and exploring which stories will have the most impact on their audience.

6.4.1 Landing Page

Our event-navigation landing page is captured in Figure 6.3. It displays stories on the news with a list of the events detected ranked by trendiness. It allows navigation from worldwide stories to country visualization, and filtering by category. The country and

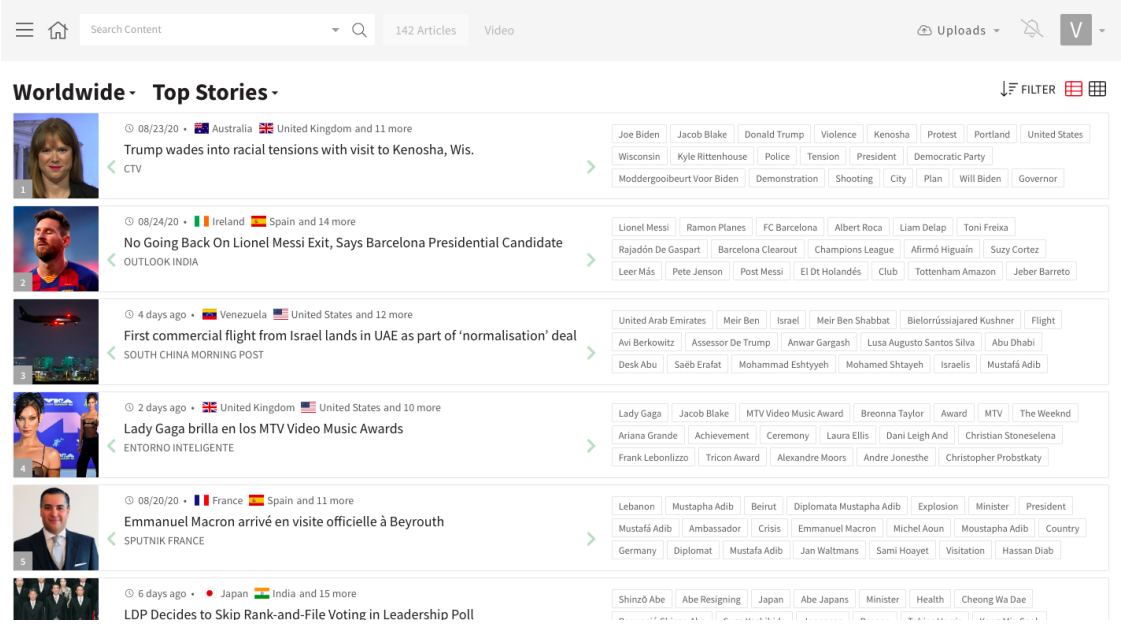


Figure 6.3: Events landing page.

category can be chosen through drop-down menus. Current categories include: *top stories*, *latest stories*, *politics*, *sports*, *entertainment*, *general news*, *business and finance*, *science and technology*, and *lifestyle and hobbies*. The menu also provides the following filtering capabilities: a) sort events by date or trending score; b) display events filtering by source: any source or only publisher related contents; and c) temporal navigation by date range.

The list of detected events is displayed behind the filters menu. It summarizes the event with the titles from the clustered articles, its most relevant entities, and an image from one of the articles. Clicking on a news story takes the user to the individual story page, where the full list of articles that were identified as being related to the topic is displayed.

6.4.2 Event Display Menu

In Figure 6.4 we present an example of the resulting event menu, which structures the information to answer the journalist W's. In the top of the menu, the event category is displayed. The title summarizing *what* happens, and the other properties: *when*, *topic*, *where* and *who* are shown behind. Next to it, the countries where articles from such event have been detected are also listed. Titles from articles clustered give context and additional information on the story, having linkage to the source page. Articles are sorted by published time in order to track the evolution of the story over time. Notice the multilinguality of the articles from the example image. Moreover, an image from one of the articles is selected in order to visually describe the event.

In the bottom, the entities in the event semantic pattern are displayed as related tags. Notice these entities are sorted according to their relevance describing the event and the size of its box represents the relevance score of each entity.

The social impact of the story is displayed throughout the Trending Chart. In this

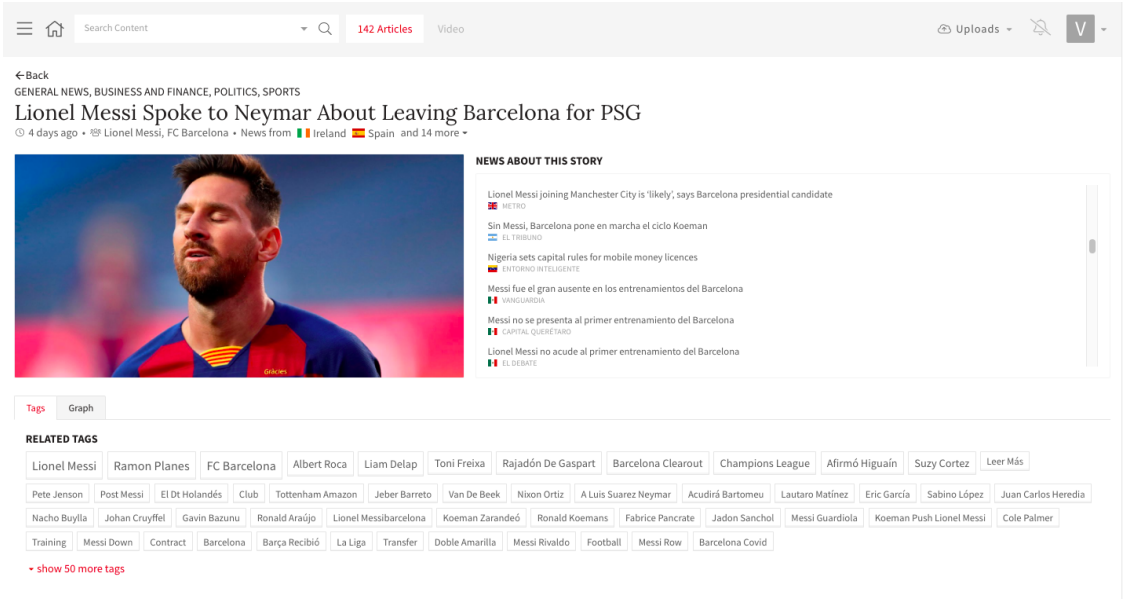


Figure 6.4: Event display menu.

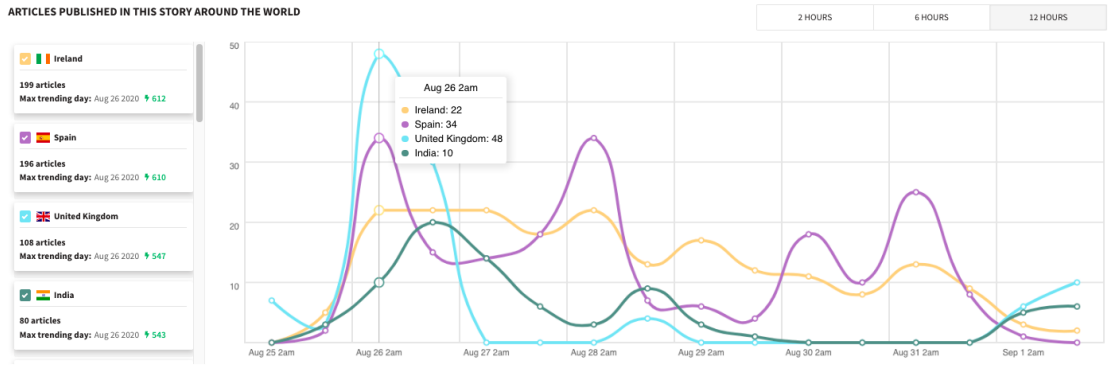


Figure 6.5: Articles published chart.

chart, the number of articles published about an story in a given country, is displayed in a graphic as the one in the example in Figure 6.5. The window of time in which to aggregate the articles can be changed from 2 hours, 6 hours and 12 hours. The displayed countries can be chose in the left menu, where additional information on the story impacted in the given country is shown: total number of articles published, day of maxim story trend and story trending score.

Lastly, when available, we aggregate all existing content about the story created by an individual publisher, as well as related content from their library. Moreover, all events and customer contents are linked through the tags. Clicking on a tag brings the user to the entity page, where all news stories and contents labeled with a given entity are displayed together. The entity page also contains a trending chart and map which displays tag trendiness on social networks.

This interface is adopted by media producers and other global media companies, which use VLX-Stories in the editorial process to identify which topics are gaining momentum,

find news related to their contents, and searching for background information on trending stories.

6.5 Conclusions

In this chapter, we presented the framework to construct the event-centric knowledge graph: VLX-Stories KG. It is an ontology-based event representation system which extracts and encodes its semantic information from aggregated news articles. The system matches unstructured text with Semantic Web resources, by exploiting Information Extraction (IE) techniques and external knowledge resources. This makes possible the multilingual linkage across events, semantic search, and the linkage to customer contents by matching entities. Moreover the ontological structure behind it facilitates event comprehension, search and navigation.

Moreover, we have presented its integration in a dashboard which displays real-time information about semantically linked events, based on aggregated multilingual news articles and linkage to customer contents. This work contributed on the UI design and backend implementation of the API. This interface is a commercial tool used by media producers and other global media companies in the editorial process to identify which topics are gaining momentum and they should be writing about, as well as the trending stories they are already covering. Moreover, the presented system allows the visualization and navigation among countries, categories, time and related entities through a friendly and intuitive dashboard.

Part III

Knowledge Graph Population

Introduction

7

Knowledge graphs (KG) play a crucial role for developing many intelligent industrial applications, like search, question answering or recommendation systems. However, when working with news contents, most of the information is dynamic and often involves unknown entities, and novel relations between entities, that are not captured in encyclopedic knowledge graphs. Detecting these out-of-knowledge-graph (OOKG) entities and facts is thus crucial when willing to provide efficient tools for news description, search and analysis. Automatically detecting, structuring and augmenting a knowledge graph with new entities and facts from text is therefore essential for constructing and maintaining knowledge graphs [87, 127, 168]. This is the task of knowledge graph population, which consists on extracting information to augment an existing data base. The two main units of data that can be missing in a knowledge graph are entities and triples, which usually appear as emerging entities, i.e. entities that are completely new or are gaining popularity; and novel facts or relations between entities.

The knowledge graph population task, usually encompasses the two main Information Extraction (IE) sub-tasks (described in Part I): (1) Entity Linking (EL) [177, 97], consisting on identifying entities from a knowledge graph in unstructured texts; and (2) Relation Extraction (RE) [219, 81], which seeks to extract semantic relations between the detected entities in the text.

In this last part of the dissertation we present a novel knowledge graph population system, by putting together the two previous parts of this work. We propose learning from aggregated news as a more reliable way to learn emerging entities and novel facts from unstructured web data than from free crawled data [133, 49]. This approach achieves both entity and triple redundancy, which allows to take more robust decisions and can be exploited by validation techniques. To do that, we extend the presented IE techniques to apply them to knowledge graph population over aggregated news articles (described in Part II). The extracted entities and facts will be used to keep up to date an industrial knowledge graph based on mass media. Ensuring the quality of the population data is thus essential. This will be handled by validation and triple classification techniques.

This part of the thesis will focus on the detection of emerging entities and valid new facts, to populate *Vilynx Knowledge Graph* (VKG). Chapter 8 describes the extension of the EL module presented in Chapter 2 to generate a dynamic EL system, capable of detecting emerging entities from aggregated news. In Chapter 9 we present validation and classification techniques applied over the RE method from Chapter 3 to discard false predictions and provide triple population with high precision.

The current chapter is organized as follows. Section 7.1 presents related work. In section 7.2, we provide an overview of the aforementioned automatic knowledge graph population

system. Finally, section 7.3 includes a summary of this part contributions.

7.1 Related Work

Recently, a lot of research effort has focused on knowledge graph population from external data sources, approaching both entity and relation discovery. Such effort is aligned with the study of IE techniques. These techniques fill the gap between machine understandable languages (e.g. RDF, OWL), used by Semantic Web technologies, and natural language, used by humans [172]. In this section we will review some of these most relevant contemporary projects for knowledge graph construction and relate it to the different IE approaches.

- **The Never Ending Language Learning (NELL)** [133]: it is a knowledge graph population system created under the *never-ending learning* paradigm. This paradigm tries to resemble the nature of human learning, in contrast with most machine learning models that learn just a single function or data model based on statistical analysis of a single data set. It is an OpenIE framework which is constantly learning new facts and correcting itself over time, as it learns to better understand natural language. This system learns both new entities and facts from large Web corpus and takes different type and domain-specific constraints to make its predictions. It is also capable of learning new constraint rules from the new facts it gathers. However, it learns relations between noun phrases, not canonical entities.

- **Textrunner** [210] **/Reverb** [55]: Textrunner is an OpenIE framework, designed for Web-scale usage, which extracts relational tuples from large corpus of data. It considers all relationships between pairs of noun phrases as candidates and then classifies each one of them as true or false. The classifier is trained with self-supervision by using a subset of the corpus automatically labeled with a set of pre-defined rules. Finally, TextRunner uses frequency statistics to determine if a fact is indeed true or not. Reverb is an extension of TextRunner which constrains relations to patterns of verbs and verb phrases which end with prepositions. Like NELL, this system extracts facts between pairs of noun phrases, as opposed to canonical knowledge graph entities.

- **Knowledge Vault** [49] : this system constructs a Web-scale probabilistic knowledge base by combining Web content with prior knowledge from existing knowledge repositories. It crawls the Web and extracts information not only from unstructured text, but also tabular data, page structures, and human annotations. They prove how combining Web content with prior structured knowledge reduces noise in the detections. Associated with each triple there is a confidence score, representing the probability of such triple being correct. In contrast with OpenIE methods, entity types and predicates detected by KV come from a fixed ontology and entities are canonical. This makes KV a structured repository of knowledge that is language independent.

- **KBPearl** [122]: this is an end-to-end system which searches to complete a knowledge graph by extracting information from a set of input documents. This system combines Open and Close IE techniques to deal with two of the main problems of

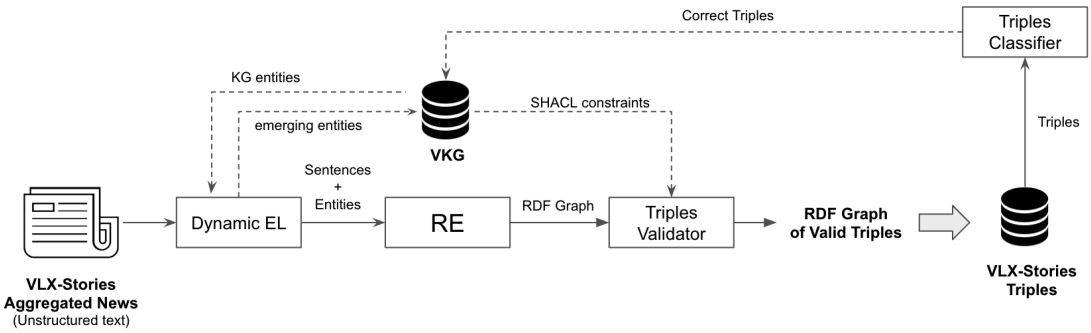


Figure 7.1: Knowledge graph population framework. The system ingests unstructured text from aggregated news and extracts an RDF graph of valid triples. These graphs are stored in a knowledge base of stories triples, and are aggregated in unique triples which are classified into correct and incorrect predictions. Correct predictions are used to automatically populate VKG. The framework is composed by four modules: Named Entity Recognition and Disambiguation (NERD), Relation Extraction (RE), a Triple Validator and a Triple Classifier.

> OpenIE methods. On one hand, the *canonicalization problem*, which consists in not solving the redundancy and ambiguity of multiple mentions resolving to the same entity. And in the other hand, the *linking problem*, which refers to the lack of linkage to ontological structures. They solve that with a method that performs a joint linking of entities and relations in a semantic graph, constructed with OpenIE. They infer new knowledge from the source text ensuring global coherence between the concepts mentioned in the documents and valid triples from the existing knowledge graph.

Our system, presented in this part of the thesis, is similar to methods like KV and KBPearl, which extract canonical entities and facts in the form of disambiguate triples. However, all mentioned methods learn from free Web crawling, while our system performs population from aggregated news. Similar approaches are taken by event-encoding systems, like ICEWS[1] and GDELT[2]. These systems extract international political incidents from news media and update their knowledge graphs online, making them applicable to real-time conflict analysis.

7.2 System Overview

This section describes the proposed end-to-end knowledge graph population framework, displayed in Figure 7.1. The system transforms unstructured text from aggregated news articles to a structured knowledge representation while detecting novel entities and triples that are OOKG. The architecture is composed by a knowledge graph and four main processing components: 1) Dynamic Entity Linking, 2) Relation Extraction, 3) Triple Validator and 4) Triple Classifier.

The input of the system are aggregated news. In this work, we define aggregated news as

[1] https://www.icews.com/
[2] https://www.gdeltproject.org/

a set of clustered articles that discuss the same event or story. These clusters are created by VLX-Stories [60] news aggregator, described in Chapter 5. This system provides unified text consisting on the aggregated articles.

The knowledge graph integrated into the current population system is the *Vilynx's*[3] *Knowledge Graph* (VKG) [59, 60], described in section 0.7. This knowledge graph contains encyclopedic knowledge, as it is constructed by merging different public knowledge resources: Freebase [25], Wikidata [197] and Wikipedia[4]. In the presented system, VKG is used to disambiguate entities in the EL module. Also, the extracted relations in the RE module are defined into VKG ontology, together with the SHACL constrains used for data validation. Finally, VKG is populated with the novel entities and facts extracted by the system.

The Dynamic EL module splits the input text, coming from the news aggregator, in sentences and detects knowledge graph entities appearing in these sentences. In this work we are extending the EL presented in Chapter 2 to detect emerging entities. Dynamic functionalities added to the EL module will allow the detection of new entities that will populate VKG. The output of this module are sentences with annotated entities.

The sentences with annotated entities are processed in the RE module. First, sentences with at least two entities are selected to produce *candidate facts*, which consist of tokenized sentences with annotated pairs of entities. For each pair of entities two candidate facts are constructed in order to consider both relational directions. Then, the deep RE model presented in Chapter 3 processes the candidate facts and extracts the expressed relation or the absence of relation. The extracted relations are expressed as RDF triples of the form $\langle subject,\ predicate,\ object \rangle$, and interconnected into an RDF graph.

The extracted RDF graph is validated with SHACL constraints, in the Validator module. During validation, we enhance results thanks to the redundancy and contextual information from aggregated news. In section 9.4.1 we give a detailed description of the constraints applied and the validation process. The output of this module is an RDF graph of valid triples. The RDF graph of triples extracted from every set of news is stored into VLX-Stories Triples knowledge base.

Finally, the triples stored in the VLX-Stories Triples knowledge base, are unified across sub-graphs into a unique graph. Prediction information from these triples is used to construct feature vectors that will be used to predict the correctness of each triple in the Triples Classifier module. If the probability of the triple being correct is above a given threshold, the triple will be selected to populate VKG. Technical details of this module are described in section 9.4.2.

7.3 Contributions

This last part of the thesis provides several contributions. First, the interconnection of all the modules previously described in Part I and Part II, providing triple extraction from aggregated news. This has allowed us to create an end-to-end knowledge graph population framework, which extracts novel facts from news events detected online.

[3] https://www.vilynx.com/
[4] https://www.wikipedia.org/

Moreover, to complete the knowledge graph population system, we extend the EL system described in Chapter 2 to detect emerging entities, as described in Chapter 8. This emerging entities provide entity search capabilities of novel things as they emerge.

Finally, to ensure high data quality, we work on triple validation and classification techniques in Chapter 9. Two independent modules are generated in order to detected potentially incorrect triples. First, a SHACL validation module is integrated into the pipeline, proceeding the RE system. It applies several constraints that will discard triples not meeting the validation rules defined in the ontology. Finally, a triple classifier is added before the knowledge graph triples ingestor, to select the triples to be added into the knowledge graph. This classifier will provide a confidence of the triple being true.

All the modules are integrated in an online knowledge graph population system, proving its performance as an industrial tool.

Emerging Entity Detection

8

Applications like entity search over news or social media allow users to precisely retrieve concise information related to entities. This expressive search mode builds on two major assets: 1) a knowledge graph with the entities of interest and 2) an entity linking system that performs document tagging by linking named entities to the entities in the knowledge graph. However, knowledge graphs are generally static and entity linking systems only recognize entities appearing in it. This limits the applicability of entity search mechanisms, that will not be able to find novel interest entities and disqualifies knowledge graphs for tasks like entity-based media monitoring, since a large portion of news inherently covers entities that just arised and thus would not be in static knowledge graphs.

Novel entities that are just starting to attract the public interest are called emerging entities. The task of emerging entity detection is crucial for any knowledge graph maintenance process, specially when using it as a base for search applications on media streams. Despite its importance for entity tagging, automatic methods for detecting and canonicalizing emerging entities have been little explored.

The system presented in Part II is capable of detecting events by clustering news articles, and represents these events by disambiguating its mentioned entities to entities in VKG. However, the representation system described in 6 will not be able to tag events with those entities missing in VKG. In order to improve these event representations, in this chapter we describe a method for detecting emerging entities from the aggregated news stories. We introduce an extension of the Event Semantic Pattern Extractor (6.2.1), to detect emerging entity candidates and populate VKG. This is done by adding an emerging entity detector system that will provide dynamics into the EL module, calling it a Dynamic Entity Linking (Dynamic EL). In this work, we will limit the system to the creation of novel entities of type person. To the best of our knowledge, this is the first system that uses the redundancy of aggregated news articles for a robust detection of emerging entities, in an online manner.

The remaining of this chapter will describe background (8.1) on emerging entities, our method to detect emerging entities (8.2), an experimental (8.3) and analytical (8.4) evaluation and conclusions (8.5).

8.1 Background

8.1.1 Emerging Entity Detection Challenges

The task of EL, related with emerging entity detection, raises different challenges resulting on four different scenarios of knowledge graph completeness regarding entities and

aliases of the entities. In [56], Farber et al. present a formalization of these challenges, which are:

1. **Known surface form, known entity**: this is the classical task of EL, when all the information is found in the knowledge graph, so that mentions and entities can be linked.
2. **Unknown surface form, known entity**: when the alias representing the surface form of the entity is missing from the knowledge graph. If not solved correctly, this challenge can result on missing linkages to known entities during the EL task, and creating duplicated entities in the emerging entity detection task.
3. **Known surface form, unknown entity**: when the surface form corresponds with an alias of another entity in the knowledge graph. This can result on linking errors because of the missed prediction of OOKG entity, and missing emerging entities that should be added to the knowledge graph.
4. **Unknown surface form, unknown entity**: Given the mention in the text, none of the alias of the entities in the knowledge graph can be matched and, hence, the mention is not linked. Also the entity to be linked to is unknown. This scenario should finish with the creation of a novel entity.

Another challenge in the creation of emerging entities, unrelated to the knowledge graph completeness, is the decision of when an unknown mentioned entity is popular enough to become an emerging entity.

8.1.2 Related Work

Typical EL systems, as the ones presented in Part I, assign a NIL label to named entities with candidate entities that have been detected with a confidence score below a certain threshold, which means the entity is out-of-knowledge graph (OOKG). However, systems that do not consider emerging entities suffer from the problems mentioned in previous sub-section (8.1.1). There is wide work on extending such simple EL systems for automatically identifying emerging entities, as this task is part of the TAC KBP [96]. In this sub-section we review emerging entity detection works related to its detection on news articles.

In [86], Hoffart et al. highlight the need of knowledge graphs to keep up with the real world's entities and suggests an alternative approach to classical EL. They argue how setting a threshold is challenging to tune and not robust enough, and study how NED methods can cope with knowledge graphs incompleteness during disambiguation, and extend them with new entities. Their approach extends a Named Entity Disambiguation (NED) model for emerging entity detection (NED-EE) by making emerging entities what they call first-class citizens. This system considers all mentions are also possible emerging entities, and adds them as an entity candidate. In order to represent emerging entities equally as knowledge graph entities, they construct keyphrases from a window of text surrounding the mention. To even further enhance the emerging entities with keywords, news articles in a close window of time (i.e. the same day) with references to the entity is assumed to be referencing the same entity. Thus, keywords extracted from nearby articles in time can be assigned to the emerging entity.

Following previous work, Wu et al. [206] presented another approach to extend NED with emerging entity detection for news articles. This new approach builds entity representations in different feature spaces (*contextual space, neural embedding space, topical space, query space and lexical space*) to decide if an entity refers to an in-KG or OOKG entity. Unlike previous works, this approach develops a novel entity classifier, module independent to EL. This work achieves better performance than previous [86] in both EL and emerging entity discover tasks. Later, [216] addressed similar issues, and analyzed the drawback of introducing an emerging entity candidate for each mention, as it results on noise due to duplicated emerging entities candidates. To address this issue, they introduce a two step process. First, a probabilistic NED yields a disambiguation score for each candidate entity. A second step searches for additional context on online Web sources, if the disambiguation score is low. This context is contrasted with the context from existing entities in the knowledge graph, leveraging thus context on the emerging entities. This results on the NED solver avoiding the noise for entities in the knowledge graph. Evaluation results display improvements with respect to [86], but Wu et al. [206] presents better F1 scores with the same dataset.

In [87], Hoffart introduces a novel approach for emerging entity detection and creation of context for this entity (keyphrases). The main requirement for adding new entities is that they should have a representation suitable for disambiguating the entity in new texts. This is done by incorporating a human in the loop, in order to grow the knowledge graph with high quality data. The user suggests a set of entity names and an initial entity description, and afterwards the system retrieves a set of candidates and keyphrases related to it that the user has to validate iteratively, until getting a good entity representation.

Another work studies the challenge of predicting when an entity appearing on the news will be added to Wikipedia [56]. First, they use the history of Wikipedia edits, combined with noun phrases extracted from news streams to analyze how news entities and its surface forms evolve in Wikipedia over time and create a dataset from it. Afterwards, they propose an emerging entity detector, based on a ML model, that detects out-of-Wikipedia entities.

8.2 Method

The presented system deals with a set of product requirements related to the industrial nature of this thesis. In one hand, we want to avoid erroneous and duplicated entities. On the other hand, to be able to search for contents related to emerging entities as soon as possible, we want to start tagging with novel entities the first time we see them. To answer this two requirements, we propose to generate an intermediate representation of entities, called *aspirants*. Aspirants will work as provisional entities, until they are validated by a human. Thus, they will be used for disambiguation and will be integrated into search algorithms as an entity itself.

The Dynamic EL module will be integrated into the event representation pipeline from Part II. Therefore, the mentions to disambiguate come from aggregated news. This implies high redundancy on the context of the extracted mentions, as they all come from different articles discussing the same topic. This allows to make the next assumption: unknown mentions with the same surface form, appearing in different articles of the same story, are referring to the same entity. Figure 8.1 displays the extended EL schema, of the

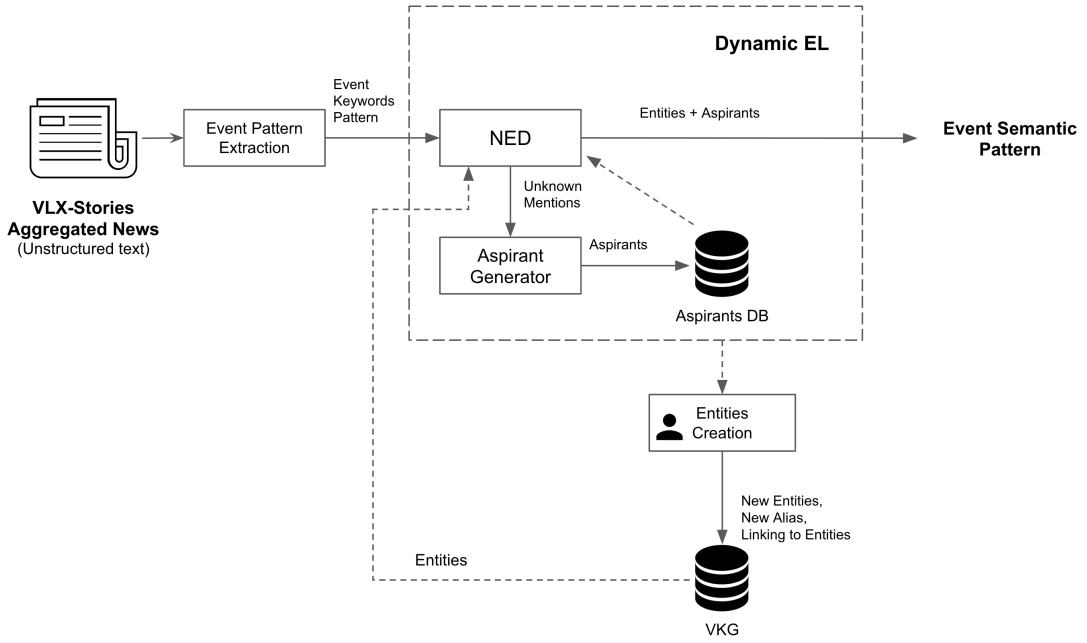


Figure 8.1: Pipeline schema of the event representation with dynamic entity linking modules. This module maintains emerging entities that refer to unknown people as they appear on the news, and integrates it into VKG.

system previously shown in Figure 6.1. Notice how the EL module has been extended to the Dynamic EL one, which includes an aspirant generation sub-module, the Aspirants DB, and the entities creation sub-module.

Next sub-sections describe the modules involved in the detection of emerging entiteis: generation of aspirants (8.2.1) and entity creation with human in the loop (8.2.2).

8.2.1 Aspirant Generation

Aspirants are defined as candidates of emerging entities. We consider aspirants all NIL entities which pass a set of conditions. Once an aspirant is detected, the system starts leveraging it for tagging, linking documents, describing new events and searching for them, as any knowledge graph entity.

Those entities that the EL system is unable to link to entities in VKG are called *unknown mentions.* These mentions are send to the aspirant generation module. In this work, we have limited the aspirant generation to keep only the unknown mentions that have been recognized as *persons* by the NER module and that are at least composed by two words (name and surname). The detection is highly robust because the emerging entities come from the previously extracted event pattern, which means the entity has appeared in a high amount of articles from different publishers, in the same context, and is thus relevant when describing the event. Aspirants created are stored as a new entry into an external DB with its metadata context (content language and sentences where it appears). Its relation to news events is stored in a secondary table. Once the aspirant is created, it is added into the Event Semantic Pattern as a descriptor, and treated equally

as other entities in the pattern on the following event representation steps (multilingual aggregation and event ontology).

Moreover, the aspirant starts collecting information, as keyphrases and other co-occurring entities in the text. This intermediate steps allow, thus, to retrieve enough information to be sure the OOKG entity is really an emerging entity and gather context about it. So, once the entity is created, it has all the needed information to be used for disambiguation.

8.2.2 Entity Creation with Human in the Loop

The creation of emerging entities requires human validation before becoming a proper concept in the knowledge graph. This validation is needed because sometimes the names detected are spelling variations (aliases) of entities already in the knowledge graph, or mistakes from the NER module. Facing these challenges, the human validator must decide if the aspirant is a detection error, an alias of an existing entity, or an emerging entity. To assist in the decision, an independent system takes care of the emerging entities by searching for entity matching suggestions in external knowledge graphs (Google Knowledge Graph (GKG), Wikipedia and Wikidata), as well as entities in VKG. Suggestion results are displayed in an internal dashboard, together with context from the sentences where the emerging entity has been seen. Thanks to previous process and auto-complete tools, the human intervention is minimal and decisions can be made very fast.

8.3 Experiments

Table 8.1: Results on Generation of Emerging Entities (EE) from Aspirants.

Country	Language	#Stories	#Deleted Entities	%EE Recovered
United States	en	282	373	80.16%
Spain	es	251	299	74.91%
Portugal	pt	115	104	85.57%
TOTAL	**-**	**648**	**776**	**78.86%**

In this section we evaluate the capacity of the aspirant generator module to detect emerging entities. It was evaluated by deleting existing entities from our VKG and testing the capacity of the system to create them again. We initially built a dataset of 648 news events detected during the week from the 1st to the 7th of January 2019. The multilingual capabilities of the system was tested by choosing events from three regions with different languages: the United States, Spain and Portugal. The dataset was generated by running the Event Semantic Pattern module (6.2.1), removing the corresponding person's entities from VKG, and extracting again the Event Semantic Pattern, expecting for the aspirant generator to re-generate the deleted entities.

As shown in Table 8.1, an average of 78.86% of the deleted entities were recovered. Some of the missing entities were composed by just one word, like *Rhianna* or *Shakira*. Our system did not detect them because it constrains person entities to be described with two words (name and surname). Other errors were caused by the similarity between entities, which are wrongly disambiguated to existing entities; e.g. when deleting the *Donald Trump* entity, the EL disambiguated to *Donald Trump Jr.* because of a perfect

match between the alias and the similar usage context.

8.4 Analytical Results

Finally, we evaluate the performance of the system deployed in production. A statistical study of the created entities and their quality was performed by analyzing data between 12th December 2018 and 15th March 2019. Table 8.2 presents the average number of emerging entities detected every day in each language. After the human validation we extracted the next metrics: 75.45% of the the detected emerging entities become new entities, 22.15% were alias of already existing entities, and 9.7% were wrong candidates because of NER errors.

Table 8.2: Statistics on Emerging Entities Detection by VLX-Stories

	EN	**ES**	**PT**	**Total**
Avg. EE detected/day	41.18	20.08	9.27	67.88

8.5 Conclusions

In this chapter we presented an emerging entity detection system. As the data generated is served to customers, we want this system to provide maximum accuracy when adding information into the knowledge graph, and to be able to start serving information related to emerging entities as soon as they appear on the news. This is solved through tagging with aspirants, a novel concept introduced by this work which consists of an intermediate representation of entities. These representations work as entities during disambiguation and are indexed by search algorithms. However, they will not be part of the knowledge graph until a human has reviewed and validated it. In the entity creation module the aspirant may be related to an already existing entity as a new alias, discarded as a mention detection mistake, or a novel entity can be created from it.

The aspirant generation module has been evaluated with a custom dataset. In this experiment, the capacity of the module to detect entities missing in the knowledge graph was tested by deleting entities of type *person* from VKG, and evaluating if the system was capable of re-generating such entities with data extracted from news stories. 78.86% of the entities were generated again. The experiment highlights the system limitations of only being able to create people entities with at least two words (name and surname). However this limitation also avoids many noisy detections, and we consider that generally only very famous people will be called with only one name and such entities should already be provided by other external knowledge graphs (i.e. Wikipedia, Wikidata...). Another point of improvement is the disambiguation step, which when deleting an entity tends to link it with other entities with the same alias. This is the challenge 3 of the EL challenges presented in 8.1.1 that must be managed from the EL prespective.

Analytical results of the system working on production show a high capacity of emerging entity detection by using the news aggregation system. An average around 68 emerging entities are detected every day. This experiment also showed that, while most of the detected aspirants are emerging entities (75.45%), there are still also a lot of missing alias (22.15%) and detection errors from the NER system (9.7%).

As future work, we will automatize the detection of alias from the aspirants. This can be easily done by using string match among multilingual alias, and using entity co-occurrences as context for disambiguation. Another relevant feature to work on is on the extension of the system to other types of entities, like '*events*, '*locations*' and '*organizations*, which are also highly descriptive of news events and often used by search engines users.

Novel Facts Validation

9

In recent years, the amount and size of knowledge structured resources has extremely grown, together with automatic methods for its population. Thus, the need for methodologies and tools able to asses the quality of these linked data resources has also increased.

Automatic methods for extracting structured knowledge from unstructured text, require Information Extraction (IE) models, like the ones presented in Part I. In particular, for extracting facts (or triples) from unstructured text, relation extraction methods are required. Over the last years, the Natural Language Processing (NLP) community has accomplished great advances regarding this IE task [30, 173]. These models achive high performance with supervised methods tested on research datasets, as the one presented in Chapter 3. However, the information extracted by these systems is still imperfect, and may compromise knowledge graphs data veracity and integrity when performing a population task.

To asses these data quality problems, the Semantic Web community has developed semantic technologies to express how the world is structured. For example, ontology languages like Web Ontology Language (OWL) represent complex knowledge and relations between things, and constraint mechanisms like Shapes Constraint Language (SHACL) specify rules and can detect data constraints violations. When building ontology-driven IE systems, these semantic techniques can be applied to asses data veracity and detect false positives before adding erroneous information into the knowledge graph.

Despite ontologies and validation tools are capable of detecting semantically wrong triples, e.g. persons being married to buildings (type constrains) or persons having several birth dates (data constraints); there are triples that are semantically correct but are still wrong. Detecting such errors is challenging, as it is not possible to detect by applying rules and more sophisticated methods are needed. A way of maintaining such high triple quality is the use of triple classification techniques. This techniques consists on training a binary classifier for predicting the validity of a triple, before adding it into the knowledge graph.

In this chapter we develop data validation and triples classification techniques to asses the quality of triples extracted from unstructured text with IE techniques (Part I). Following previous work on aggregating news articles (Part II), we are going to validate the quality of triples extracted from articles clusters. As described in Part II, aggregated news are clusters of news articles describing the same story. While web-based news aggregators such as *Google News* or *Yahoo! News* present these events with headlines and short descriptions, we aim towards presenting this information as relational facts that can facilitate relational queries. As shown in Figure 9.1, the system ingests unstructured text from these news stories as input and produces an RDF graph as output, completing all the

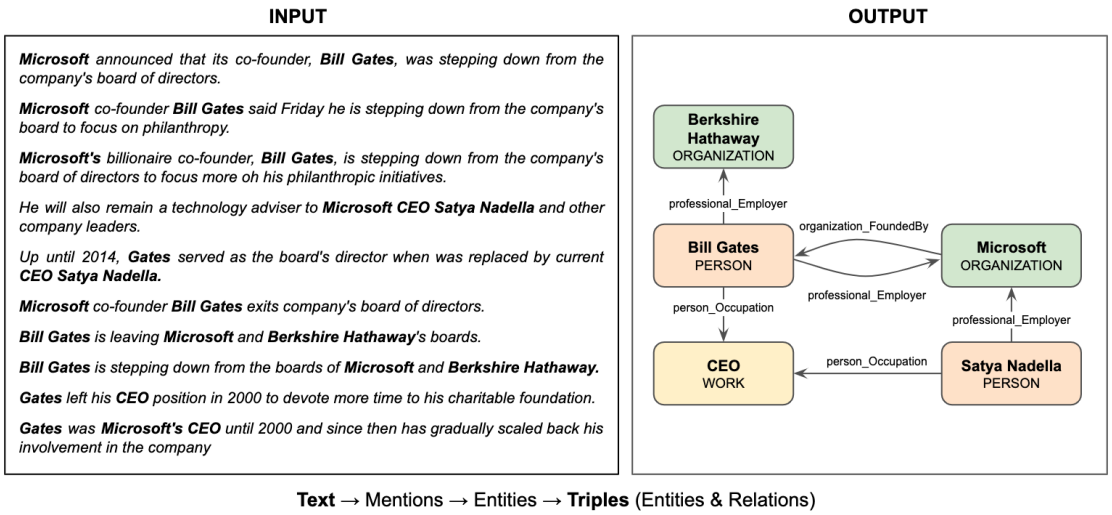


Figure 9.1: Example of graph constructed from sentences from aggregated news articles.

way from unstructured text found in the Web to a knowledge structured representation of data.

In the example from Figure 9.1, we can see how many of the sentences in the input text are expressing the same relations, e.g. sentences "*Microsoft announced that its co-founder, Bill Gates..*", "*Microsoft's billionaire co-founder, Bill Gates...*", and "*Microsoft co-founder Bill Gates said...*" can all be synthesized with the triple ⟨*Microsoft, FoundedBy, Bill Gates*⟩. The validation system takes advantage of this redundancy, as well as other extracted triples, to detect contradicting information while verifying against our ontology and the knowledge graph data. The triple classifier will also take advantage of such redundancies. Instead of basing the triple validity prediction on the contextual probability, the classifier will use features based on the redundancies given by the aggregated news.

The contributions of this chapter can be summarized as: a) The addition of a validation module on top of a relation extraction module, which exploits the context and redundancy from aggregated news. We show how this validation highly increases overall data quality on the new AggregatedNewsRE[1] dataset presented. b) The construction of a triple classifier to ensure the automatic population offers high precision.

The remaining of the chapter is structured as follows: an overview of the integration of the triple validation and triple classification modules in the whole knowledge graph population framework is described in section 9.1; background on semantic constrains is presented in section 9.2. In section9.3 we present related work on RDF validation and triple validation. Following, our proposed triple validation and the triple classification methods are presented in 9.4. Section 9.5 provides experimental evaluation on both systems, and section 9.6 presents analytical results of the system on production. Finally, we provide some conclusions in section 9.7.

[1] https://figshare.com/articles/dataset/AggregatedNewsRE_Dataset/12850682

9.1 System Integration Overview

In this chapter we describe the triple validator and triple classifier modules. These two modules integrate into the whole knowledge graph population framework from Figure 7.1. Notice how the first module discards semantically incorrect triples while the second one uses redundant predictions to avoid adding incorrect triples into the knowledge graph. In this section we review the integration of these two modules into the whole population system.

As can be seen in Figure 7.1, the triples validator ingests the triples outcoming the RE module, which are represented as an RDF graph. This module validates the graph by applying the SHACL constraints defined in the knowledge graph, and outputs an RDF graph of valid triples. Such triples are stored as RDF graphs for each event in VLX-Stories Triples knowledge base.

Finally, all triples from the RDF graphs in VLX-Stories Triples are aggregated into unique triples. Information from all predictions is gathered and feature vectors are constructed from it. These vectors are feed into the triples classifier, which estimates the probability of a triple being true. If this probability is above a given threshold, the triple is added into VKG.

9.2 Background

When building knowledge graphs from unstructured or semi-structured data, information extracted is specially vulnerable to quality issues [107]. In this section we overview the main SHACL constraints used in this work (9.2.1).

9.2.1 Constraints Overview

We divided the validation rules applied in two main groups: *type constraints*, where validation is based on rules from the pre-defined ontology concerning the entity-types a relation can connect; and *data constraints*, where validation relies on data from other triples in the KG.

- **Type Constraints**: When defining an ontology, *domain* and *ranges* are associated to the different kinds of relations. These properties describe if a relation can link a subject to an object, based on its associated type classes. The *domain* defines the types of entities which can have certain property, while the *range* defines the entity types which can work as an object. Domain and range properties also apply to types sub-classes defined in the ontology hierarchy. As an example, if the relation "*FoundedBy*" is applied from a root domain "*Organization*" to a root range "*Person*", this means entities with types or sub-types of this domain and range can be linked by this property. However, if we restrict the relation "*MemberOfSportsTeam*" to the domain "*sportsPerson*" and range "*sportOrganization*", only the entities with these sub-types will be linked by this relation. For all relations in our ontology we defined their respective domains and ranges, which will be used for validation.

  Notice that when applying this rule we will discard false positives, but if we are

missing entity-types relations in the KG, we will also discard some true positives. For example, we may know some entity is type "*Person*", but if we do not have the association of this entity with the sub-type "*Politician*", we will discard triples of this entity involving the relation "*MemberOfPoliticalPary*" or "*HeadOfGovernment*". While this will cause a decrease in recall, it is also an indicator of missing entity-type relations that should be populated. Nevertheless, this problem is currently not analyzed, and in this work these triples will be discarded.

- **Data Constraints**: We define two kinds of data constraints: *cardinality* and *disjoint*. Cardinality constrains refer to the number of times a property can be assigned to an entity of a given domain. For example, an entity of type "*Person*" can have at most one "*BirthDate*". This constraint can also be applied considering time range statements, to guarantee e.g. that a country does not have two presidents at the same time. Disjoint rules guarantee that entities have to be disassociated for a set of properties. For example, if two entities are known to be related as *Siblings*, they can not be associated as *Parent* or *Child*. We apply this kind of restriction to relations concerning the *Person* domain in connection to family relation properties like *Parent*, *Child*, *Sibling* and *Partner*. Moreover, we consider inverse predicates when applying these constraints.

9.3 Related Work

9.3.1 RDF Validation

When constructing a knowledge graph, its data is only valuable if it is accurate and without contradictions. Requirements for evaluating data quality may differ across communities, fields, and applications, but nearly all systems require some form of data validation. Following this approach, different works analyzed the consequences of errors in knowledge graphs and established recommendations [91, 190]. The detection of inconsistencies and errors in public knowledge graphs has also become the subject of various studies during the past years. Many works analyzed errors in public semantic resources like DBPedia and Wikidata, and proposed automatic methods that combine statistics and validation tools to detect them. For example, in [191], Topper et al. enrich the DBpedia ontology by using statistical methods to detect inconsistencies during its population. Another interesting work by Spahiu et al. [185] presents a method that extracts data-driven ontology patterns and statistics, and detects data quality issues across different versions of the data by means of semantic constraints.

When validating graphs, there are different RDF validation languages to define these constraints, but shape approaches like ShEx [68], SHACL [105] and ReSh [166] are the ones receiving the greatest community support and advanced features [190]. In particular, SHACL (Shapes Constraint Language), has become the latest standard and the W3C recommended system for validation of RDF graphs.

Following these recommendations and aiming at a high level of data integrity in our knowledge graph, in this work we will describe the integration of a SHACL validation module into our knowledge graph population system.

9.3.2 Triple Classification

As knowledge graphs usually suffer from incompleteness, an important research topic is to predict the missing connections in the knowledge graph. The most basic task to solve this problem is triple classification, which tries to estimate the veracity, or the degree of truth, of an unknown triple. This systems are generally based on embeddings [48], as it is also a common technique to evaluate the quality of an embedding model [184].

Some IE systems have already integrated triple classifiers at the end of the pipeline to detect invalid triples. For example, Distiawan et al. [46] present an end-to-end IE system for knowledge graph population, based on a neural encoder-decoder model, complemented by a triple classifier model. The triple classifier is trained by using entity embeddings computed with the knowledge graph embeddings model, TransE [26] and is used to filter invalid triples generated by the neural relation extraction model, improving the whole system precision.

In this dissertation, we propose a new triple classification method. Our method, instead of using embeddings to measure the triple coherence, uses the redundancy of many predictions from a RE module and extracts features from the origin sentences that will be useful to validate the confidence of the triple predictions.

9.4 Method

9.4.1 RDF Validation

To enhance extracted triples quality, we propose knowledge graph population on aggregated news over free crawled data, and a validation method that exploits the redundancy on this information. On one hand, the fact that articles come from verified sources and have been clustered on news story events, increases the trustfulness of the text and ensures that the content from which we learn is relevant. On the other hand, the aggregated articles talk about the same agents and events, adding redundancy and context to the predictions.

In this section we describe the validation preformed to an RDF graph extracted from an aggregated news content. We will start presenting the nomenclature used, and continue with the algorithm.

An RDF graph G is constructed by a finite set of triples $\mathbf{t} = [t_0, ..., t_n]$, where $0 \leq n$. Triples are of the form (s, p, o), where s is the subject, p the predicate and o the object. s and o are the *nodes* elements in the graph G, and p the *edge*. Particularly, given a set of RDF triples $\mathbf{t}_{AN}$, extracted from an aggregated news (AN) content, and composing an RDF graph G_{AN}, our triple validator follows the methodology described in algorithm 1.

9.4.2 Triple Classification

The approach taken to verify the confidence of the triple prediction is inspired by the triple classification method, commonly used to evaluate entity embeddings [184]. This method consists on training a classifier which estimates the probability of a given triple being true. Nevertheless, our estimation is not going to be based on context and embed-

Algorithm 1 Triple validation algorithm

1: Repeated triples in G_{AN} are merged in a graph of unique triples $\hat{G}_{AN}$, where $\hat{G}_{AN} \leq G_{AN}$.
2: The occurrence count for each unique triple is stored in a counter $\mathbf{c} = [c_{\hat{t}_0}, ..., c_{\hat{t}_m}]$, where $c_{\hat{t}_j}$ is an integer ≥ 1 with the number of occurrences of a unique triple $\hat{t}_j$.
3: A second graph (G_{KG}) is constructed with all KG triples from entities appearing in the same aggregated news content.
4: $\hat{G}_{AN}$ is extended with G_{KG}, being $G = \hat{G}_{AN} \cap G_{KG}$.
5: SHACL constraints are applied to G.
6: The SHACL validator outputs a set of a valid triple $\mathbf{t}_v$, invalid triples by type $\mathbf{t}_{it}$ and a list of alternative sets of incompatible triples by data constraints $\mathbf{T}_d = [\mathbf{t}_{d_1}, ..., \mathbf{t}_{d_k}]$ where each set $\mathbf{t}_{d_l}$ is composed by a valid triple t_{vd} followed by the triple that would be incompatible with the previous one t_{id}.
7: **if** triples are invalidated by type constraints ($\mathbf{t}_{it}$) **then**
8: Discard triple
9: **end if**
10: **for** each set of incompatible triples by data constraints ($\mathbf{t}_{d_l}$) **do**
11: **if** triple $t_{vd_l} \in G_{KG}$ **then**
12: Correct Set. The invalid triple (t_{id_l}) in the set is discarded.
13: **else**
14: **if** $c_{\hat{t}_{vd_l}} > c_{\hat{t}_{id_l}} + \alpha$, (being $\alpha \in \mathbb{R}$ and $\alpha \geq 0$), **then**
15: Correct Set. Discard invalid triple t_{id_l}.
16: **else**
17: Incorrect Set. Discard all triples in $\mathbf{t}_{dl}$
18: **end if**
19: **end if**
20: **end for**
21: Final output consists in an RDF graph of valid and unique triples extracted from the aggregated news content, $\hat{G}_{AN_v}$

dings features, as it usually does when willing to evaluate embeddings. As the prediction has been done through a RE model, we generate a feature vector based on the sentence that originated the prediction, the trained model and the prediction confidence. As RE is applied on top of aggregated news, we find several sentences expressing the same relations between concepts, and thus, we expect a lot of equivalent triples. We use this redundancy as a metric of confidence, as we believe it is more probable that the estimation is correct if it has been predicted several times in different sentences.

First, all triples stored in VLX-Stories Triples will be unified into unique triples, i.e. multiple equivalent predictions are unified to one. Then, the information related to all predictions that lead to the same unique triple is going to be used to construct feature vectors. After analyzing several RE prediction results, we observed different variables that make the predictions less reliable. For example, when there are a lot of entities in a sentence, the relation extractor tends to fail by associating the same relation among all entities. Another variable that influences on the prediction, is the distance between entities: the further the entities are, less reliable is the prediction. We translate these variables into features that construct the feature vectors used to evaluate the confidence of a triple prediction. The features that compose the feature vector used for classification

are:

- **Total sentences**: it is the total number of sentences from which a triple has been extracted.
- **Total events:** it is the total number of independent events or stories where a triple has been predicted.
- **Number of times valid**: it is the number of times the SHACL triple validator considered it valid.
- **Number of times invalid**: it is the number of time the SHACL triple validator considered it invalid.
- **Class precision**: corresponds to the RE precision when estimating a the predicate of the triple under evaluation.
- **Distance between entities**: to encode the distance between entities for all the sentences that originated a triple, we construct a feature of dimension 4. Each position encodes the number of sentences where the entities had a distance among some ranges. First position goes from 0 to 10, second goes from 11 to 20 , the third from 21 to 30, and the last one over 30 words apart.
- **Confidence range**: similar to the previous feature, in this feature we want to encode the prediction confidence for all the sentences that originated a given triple. It will be done with a feature of dimension 4, where the first position represents the number of predictions with confidence between 1 and 0.96, the second position includes the confidence range between 0.95 and 0-91, the third goes from 0.90 until 0.85, and the forth one represents predictions with confidence below 0.85.
- **Number of concepts in the sentence**: last feature represents the number of concepts in a sentence. As previous two features, we encode counts of predictions in ranges. It is going to be represented with a 5 dimensional vector, where the first position is the count of sentences with 2 or 3 entities, second position represents sentences with 4 to 5 entities, third from 6 to 7, forth from 8 to 9, and finally the fifth position is the count of sentences with over 9 entities that lead to a prediction.

All the described features are encoded for each unique triple predicted in an 18 positions feature vector. These vectors are sent to a binary classifier that predicts the validity of the triple. Any binary classifier can be used for this task.

9.5 Experiments

In this section we present the experimental evaluation of the two methods proposed in this chapter to asses triple quality. The aim of these experiments is to evaluate the proposed methods for detecting invalid triples. First experiment (9.5.1) measures the capabilities of detecting inconsistent triples using SHACL constraints, and how it can be improved by exploding data redundancies from the aggregated news. Second experiment (9.5.2) is to evaluate the quality of the triple classifier. We test different classical machine learning classifiers and compare its results.

9.5.1 RDF Validation

The effects of each step from the validation algorithm presented in subsection 9.4.1 are analyzed here. We want to see the capabilities of this module to detect erroneous triples and evaluate validation in the aggregated news context.

9.5.1.1 Dataset

We generated a manually annotated corpus of candidate facts extracted from aggregated news collected by VLX-Stories (described in Part II), which we call AggregatedNewsRE. This dataset is used to evaluate the contribution of the presented RDF validation module and analyze the applied constraints. Sentences from aggregated news were annotated by our EL module (presented in Chapter 2), and candidate facts were constructed for each sentence where entity pairs were identified. After this pre-processing, the relations in this candidate facts were manually annotated by one expert annotator. The resulting dataset contains a total of 11 aggregated news stories and 400 candidate facts. Diverse topics were selected for these news, in order to cover different kinds of relations. The final aggregated news corpus includes 17 from the 27 relations in the TypeRE dataset. Table 9.1 shows the AggregatedNewsRE dataset metrics.

Table 9.1: Metrics of the AggregatedNewsRE dataset.

Dataset	#Total	#Relations	#Entities	#Aggregated News
AggregatedNewsRE	400	17	91	11

9.5.1.2 Results

We extract triples for all the candidate facts in the AggregatedNewsRE dataset, using the RE model introduced in Chapter 3, $\text{BERT}_{\text{EM+TM}}$. On top of these results we perform three different levels of RDF validation, that we analyze in Table 9.2. Notice the performance on the base result is low in comparison to scores presented in Table 3.3. This is because the sentences in the TypeRE dataset, used to train the model, are from Wikipedia articles, while sentences in AggregatedNewsRE dataset are from news articles, where language expressions follow a different distribution.

Table 9.2: Comparison on the validation contribution when using contextual information of all RDF graph extracted from aggregated news (AN). We compare the output from the RE model (Base), type constraints (Type), all constraints validated against our KG (Type+Data) , and all constraints validated against VKG and the triples in the RDF graph extracted from the aggregated news (Type+Data in AN).

	Precision	Recall	F1	Accuracy
Base	54.5	85.5	66.6	62.3
Type	60.0	85.1	70.4	67.6
Type+Data	62.8	85.1	72.3	70.0
Type+Data in AN	**70.1**	**81.7**	**75.5**	**75.0**

Our experiments compare different levels of RDF validations. First, we apply Type Constraints, which discarded 35 triples and improved precision by a 5.5%. Second, we test the validation of each individual triple using the SHACL constraints. This applies

both Type and Data Constraints, and discards a total of 50 triples, increasing precision an 8.3%. Finally, we validate the RDF graph extracted for each group of aggregated news. This last validations uses the redundant information from the aggregated news, discarding a total of 95 triples and improving precision by a 15.6%, with respect to the baseline. For this last experiment, α was set to 2. As can be seen, the main effect of validation is an increase in precision, thanks to the detection of false positives. As expected, recall is lowered down by the Type Constraint due to incomplete entity-type information. When the validation process uses all aggregated news RDF graph, some true positives are discarded due to contradictions between extracted triples. Nevertheless, notice that only a 3.8% of recall is lost, while accuracy increases 12.7%.

9.5.2 Triple Classification

The triple classification model described in section 9.4.2 is trained and analyzed in this subsection. This evaluation aims to compare different kinds of classifiers, analyze the impact of each feature on the final decision and tune parameters.

9.5.2.1 Dataset

To train and evaluate the triple classifier, we constructed a dataset with triples extracted from the aggregated news articles, processed by VLX-Stories (described in Part II). Notice we did not use the AggregatedNewsRE dataset, used in previous subsection, because we needed a larger amount of sentences and variability to train the triple classifier model. To generate the dataset, repeated triples across contents were aggregated into unique triples. Accompanying each triple, all the sentences that originated it are stored. Using the triples data and the origin sentences, the triples were manually annotated by three persons with valid and invalid labels. In case of label disagreement, the most voted one was chosen. The final dataset has a total of 866 unique triples, consisting on 500 valid triples and 366 invalid triples.

9.5.2.2 Training

For each triple in the dataset we construct a feature vector consisting on the features described in 9.4.2. Samples in the dataset are divided into train and test subsets, with a partition of 70% and 30% correspondingly. With these partitions we will train four different classifiers: linear SVM, gradient boosting, random forest and bagging classifier. The classifiers are trained through the machine learning tools from Scikit-learn [146] library.

9.5.2.3 Results

We classify all triples in the test partition as valid or invalid with the four classifiers. Final results on the triple classification are displayed in Table 9.3. Maximum accuracy is with the gradient boosting classifier, reaching a 73.8% and a ROC AUC of 81%.

In order to provide knowledge graph population with high precision, we want to set a high probability threshold of the triple being correct. To choose the threshold and the classifier that best fits our purpose, we also analyzed the precision/recall decay when setting a threshold into the true class probability value. We changed the threshold from

Table 9.3: Comparison of triple classification results with different classifier models.

	Precision	Recall	F1	Accuracy	AUC ROC	AUC P/R
SVM Linear	82.2	60.6	69.8	67.7	78.9	0.835
Gradient Boosting	81.1	**75.0**	**77.9**	**73.8**	**81.0**	**0.85**
Random Forest	84.2	63.1	72.1	70.0	78.2	0.83
Bagging	**84.3**	0.669	74.6	71.9	78.5	84.8

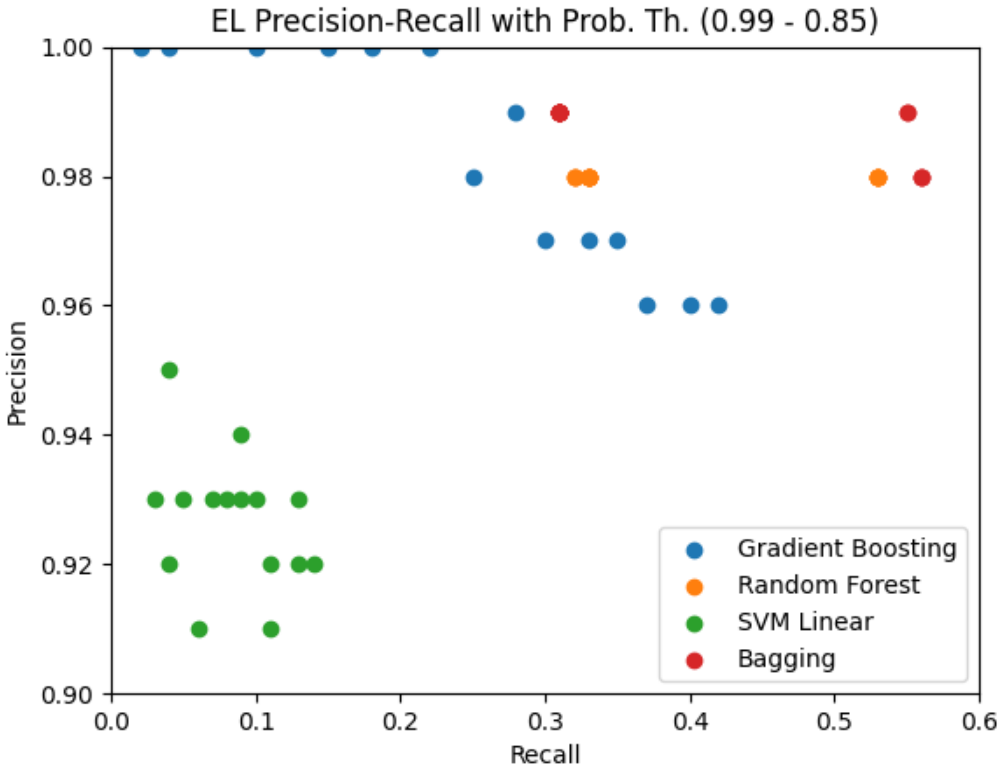


Figure 9.2: Precision Recall Values for the tested classifiers when varying the total score threshold from 0.89 to 0.99 in 0.1 steps.

0.99 to 0.85 with steps of 0.01 and checked its effect on the four classifiers. Results are presented in Figure 9.2. We observe Linear SVM decays fast on precision, while recall is very low. The other classifiers provide better results, specially random forest and bagging classifiers, which are able to provide high precision (98%) with a recall above 50%, with a threshold between 0.89 and 0.85.

9.6 Analytical Results

In this section we evaluate the performance of the system in production for the two modules presented in this chapter.

9.6.1 RDF Validation

In this subsection we study the quantity and quality of the generated triples on the online knowledge graph population system under study. We analyze triples extracted from 171 aggregated news, collected during a period of time of 24h. From these news stories 706 triples have been obtained, setting an average of 4.12 triples/content. However, if we aggregate repeated triples extracted from the same content, we have a total of 447 triples. These values show high redundancy on these data.

The final population system not only validates triples with SHACL constraints, but also filters out triples with a prediction confidence lower than α=0.85. This threshold has been chosen to prioritize precision over recall in order to boost data quality. From the 447 triples extracted, 29.98% are valid, while 70.02% are invalid. Among the invalid triples, 56.23% were discarded by the confidence threshold, 35.46% because of type constraints, and 3.68% for data constraints. From the remaining 134 valid triples: 72.5% are new. We manually evaluated these new triples and stated that 88.6% of them are correct.

9.6.2 Triple Classification

We analyze the final knowledge graph population results on the production system. For the triples classifier, we use the model trained with the bagging classifier, setting a probability threshold of 0.85. According to previous experiments, this threshold provides a precision of 98% with a recall of 56%.

The knowledge graph population module is executed once a day, selecting which triples extracted from VLX-Stories are sent to VKG for ingestion. We analyze results from an execution from the 13th of May 2020: 113 unique triples, extracted from unstructured text, are sent to the VKG. From these triples, only the 57% are finally ingested by VKG, the other ones are discarded because they were already in the knowledge graph. We manually analyzed the ingested triples and we detected 10 wrong triples, stating that a 84.6% of the total triples ingested are correct.

Additionally, we analyzed the incorrect triples in search of patterns. We noticed most of them having a good relation prediction, but being wrong because an incorrect entity disambiguation. We believe this could be solved by introducing knowledge graph embeddings on the classification, that would provide context, which will be addressed in future work.

9.7 Conclusions

In this chapter, we explored opportunities in the intersection between NLP and Semantic technologies. We demonstrated how combining both modalities can provide improved data quality. The contributions presented in this chapter are focused on the validation and quality enhancement of the triples extracted to provide a high precision knowledge graph population.

First contribution is based on the usage of semantic technologies for triple validation. On top of an IE system, we have built a SHACL validator module that ensures coherence and data integrity to the output RDF graph. This module enforces restrictions on relations to maintain a high level of overall data quality. The novelty in this module resides in

exploiting context and redundancy from the whole RDF graph extracted from aggregated news. Finally, we provide metrics on the system performance. Thanks to the validation module the system increases precision by a 15.6% over the baseline, discarding most of the erroneous triples.

Second contribution consists on a triple classifier module. This classifier selects those most probably correct triples to be added into the knowledge graph, preventing it from growing with noisy data. This module aggregates triples extracted from multiple aggregated news and classifies them into correct and incorrect triples. The probability of the triple of being correct is compared against a threshold to decide if the triple should be ingested by the knowledge graph, or not. Unlike common triple classifiers, which are based on entity embeddings, our method constructs feature vectors based on attributes extracted from the sentences that originated the prediction, and the classifier itself. The extracted probability can be understood as a confidence score based on redundant triple predictions. Experimental results show a classification accuracy of up to 73.8% with a gradient boosting classifier. According to our results, we can warranty a precision of 98% with a recall of 58%, when using a bagging classifier and setting the probability threshold at 0.85. Finally, all the modules are integrated into the production system, which daily detects over 100 triples, which are sent to VKG for ingestion.

The two models presented in this chapter can be used independently, but they complement each other when applied together, ensuring a greater data quality. In one hand, the SHACL validator module ensures data sent to VKG is coherent with the type constraints and does not contradict other information in the knowledge graph or in the other triple predictions. In the other hand, the triple classifier provides a confidence filtering. This system is, thus, capable to detect incorrect predictions that are valid according to SHACL constraints.

As future work, we plan on extending the triple classifier with entity embedding models, in order to combine the prediction metrics used with contextual features. Moreover, if we can collect a bigger dataset, neural models should be tested as classifiers.

Conclusions

Knowledge graphs are a popular technology for organizing and representing structured information, but existing methods for constructing and populating them have limitations, particularly in terms of dynamics, timeliness and data integrity. By leveraging the wealth of information contained in news streams, it is possible to overcome these limitations and create knowledge graphs that are more comprehensive, dynamic and ensure quality data. In this thesis, **we explored the industrial opportunities offered by the automatic extraction of information from news streams to create and populate knowledge graphs**. This chapter concludes the doctoral dissertation by summarising the key research findings in relation to the research questions. We discuss the value of our contributions and review the limitations of the study.

At the beginning of the thesis we stated the principal research question of how to generate a dynamic knowledge graph that updates according to news changes. To answer this question we studied Information Extraction (IE) methods, Topic Detection and Tracking (TDT), event representation tools, and semantic technologies. The combination of all these technologies resulted on a novel end-to-end system that provides efficient population of knowledge graphs from aggregated news streams. As far as we are concerned, it is the first knowledge graph population system based on aggregated news articles. Moreover, we present a novel holistic understanding of multimodal data from the news for a better entity extraction. The findings of this study demonstrate how both aggregated news and multimodal information provide redundancy that can be exploded to provide more reliable knowledge. The presented system is also an industrial tool that maintains an event knowledge graph (VLX-Stories KG) and a media oriented knowledge graph (VKG) with real world updates. This industrial thesis has been developed in Vilynx and all the contributions have been deployed as industrial tools on production systems, providing services to several media companies.

The presented system is composed by three main modules, in which we structured the thesis: IE models (part I), event detection and representation (part II) and triple validation (part III). All together, they compose the following knowledge graph population from news streams pipeline:

- First, a news aggregator, called VLX-Stories, parses news articles and clusters them according to the event they discuss (Chapter 5).

- Afterwards, IE techniques are used to detect the entities appearing in the aggregated articles (Chapter 2) and the relations between them (Chapter 3). Moreover, the entity extraction model explodes multimodal data in news articles through an holistic understanding of the information.

- This system is extended with an emerging entity detector module, which populates the knowledge graph with novel entities appearing in the news (Chapter 8).
- The relations extracted between the entities are validated with semantic technologies, which also take advantage of the redundancy provided by the fact of extracting information from aggregated articles (Chapter 9). These validated triples populate an event-centric knowledge graph, called VLX-Stories KG.
- The event discussed in the aggregated articles is synthesized by extracting the "*who*", "*what*", "*where*", "*when*" and the main "*topic*" under discussion. This information is also stored into VLX-Stories KG through the proposed event ontology (Chapter 6).
- Finally, the triples inferred are used to update the relations in a media oriented knowledge graph, called Vilynx Knowledge Graph (VKG). In order to ensure the quality of the data added into VKG a triple classifier is added to discard incorrect triples (Chapter 9).

Each part had its own research questions and contributions. In the following paragraphs we summarize the findings and contributions of these three parts and chapters of the thesis.

Part I studied IE methods to transform unstructured data to structured knowledge. To this end, we analyzed Entity Linking (EL) and Relation Extraction (RE) systems, as we explain next.

Chapter 2 focused on EL methods, and we created our own holistic model. We stated the research question of if it was possible to enhance entity extraction by using multimodal information. As news are composed by multimodal contents (audio, images, videos and text) we explored the particular challenges of holistic EL systems and built a model that combines different forms of multimodal input data and extraction methods. During the experimental evaluation, we analyzed the system performance with different classifiers and simulated the addition of visual recognition detections. The final best EL model got a performance of an 82% F1 score when adding visual data, which represent a 5% improvement from the base model. With this experiment **we showed multimodal information helps enhance the extraction of entities**. The second question stated was the applicability of the entities extracted from video descriptions to tag video contents. This question was important for the industrial deployment of our entity extraction model and bringing our work to products. We performed a human rating evaluation of the tags extracted from 1,400 video descriptions from the YouTube-8M [1] dataset. Experimental results showed a tagging accuracy of 80.87%, and the capability of capturing relevant non-visual concepts, descriptive of the contents. We conclude that **tags extracted from the descriptions accompanying video contents provide rich information, useful for indexing multimodal contents**.

Chapter 3 focused on the second part of the IE system, the RE module. In this chapter we analyzed different RE methods and studied the contribution of semantic information (i.e. entity types) when performing this task. For that, we introduced the concept of *Type Markers*, and added them as a new token into the BERT_{EM} RE model from Soares et al. [183]. Experiments proved a 3.3% F1 performance increase with respect to the baseline, on the well known TACRED [220] dataset, and a 2.2% increase on the TypeRE

[61] dataset. We can state, thus, that **semantic knowledge does help in improving RE predictions**.

Part II of the dissertation studied the detection and representation of events from news articles. The goal of this part was the construction of an online system capable of automatically detecting news events and representing them in an structured way. To this end, we developed VLX-Stories, a system under exploitation that aggregates news articles from news feeds and represents its contents as a knowledge graph. We distributed the work into two main problems: event detection and event representation. Next, we explain our approach and contributions solving these two problems.

Chapter 5 focused on event detection. We studied topic modeling and TDT literature to create our own news aggregation system based on named entities. This system creates clusters of articles in a language and regional base. **Considering each group of news articles an event, we completed the objective and research question of automatically detecting news events.** Evaluation of the news aggregation system was performed with the UCI-ML News Aggregator Dataset [45]. Experimental results showed a precision of 81.58% detecting events, and an F1 score of 72.32% when classifying news articles into story groups. Moreover, these news aggregation has been brought to production. This industrial deployment has been one big challenge of our work, as it required to work on near-real-time and to be scalable with the increasing amount of data. To reduce the computational load we restricted the article clustering into countries and categories, reducing the amount of data to be processed in each job. Moreover, we used multi-threading and parallel processing in order to accelerate the data processing time. **The final news aggregation system processes an average of 17,296 articles/day, detecting over 350 worldwide events/day from seven different countries (United States, Spain, Canada, Australia, Ireland, United Kingdom and Portugal) and three languages (English, Spanish and Portuguese)**.

In Chapter 6 we worked on the second part of VLX-Stories, which is event representation. In this chapter we wanted to solve the problem of synthesizing relevant news information to quickly explore and understand and event. To this end, we studied the application of IE tools for event representation and analyzed different semantic models to structure news information. Our final method proposed uses a Named Entity Disambiguation (NED) model to extract the main entities representing a news story, which we call the *semantic pattern*. Afterwards, these entities are used to unify event clusters into world wide stories. Final events are represented by means of an ontology inspired on the Ws from journalism, and it is stored into VLX-Stories KG. The quality of the entities extracted for each event was manually analyzed, reaching a F1 score of 90%. **This tool is used by media publishers to accelerate the editorial process, and it is helping them on content discovery, search and to explore stories impact**. In this thesis we also contributed with the design of the interface and APIs to serve this information and we added a section describing it (section 6.4).

Finally, in Part III, we stated the question of automatically detecting world changes. To this end, we put together previous pieces of the worked developed, and studied the automatic population of a knowledge graph with novel entities and facts. Additionally, as an industrial thesis, data quality is one of the main concerns of our system. That is why we have put research efforts on the validation of triples. We distributed this work in two chapters, one focused on the automatic detection of novel entities, and the second

one dedicated to novel fact validation, as explained in the next paragraphs.

Chapter 8 introduces an extension of the EL pipeline to provide emerging entity detection. The system automatically creates an intermediate representation of entities, called "*aspirants*", which, after human validation, will become knowledge graph entities or new alias. The emerging entities detection capabilities were tested by deleting entities from the knowledge graph and evaluating if they were created again with data from news stories. Results concluded with almost 80% of the entities deleted being created again. Current system is deployed on production, generating an average of 68 emerging entities a day. **Research contributions of this chapter are based on the introduction of "*aspirants*", as a novel entity representation to dynamically integrate emerging entities into production systems, while ensuring a clean knowledge graph population with novel entities**.

Chapter 9 completed the knowledge graph population by validating triples inferred by the RE model, before ingesting them into the knowledge graph. In this chapter we studied two different kinds of data quality verification: RDF validation and triple classification. These two methods are complementary: RDF validation applies constraints to prevent inconsistent data to be added into the knowledge graph, while triple classification predicts the probability of a triple to be true, regardless of the ontology consistency. In this chapter we explored the benefits of using semantic technologies on top of NLP models (e.g. RE model). **We applied SHACL constraints on the triples extracted from aggregated news articles, and experimental results showed a 15.6% precision improvement.** Afterwards, we apply triple classification to predict the veracity of the valid triples. We construct feature vectors for each unique triple predicted by unifying data extracted from many equivalent predictions. These vectors are fed to a binary classifier that predicts the trustability of the triple. We tested several classical classifiers, and the best performing one is gradient boosting, with an accuracy of 73.8%. In the production integration, we added a confidence threshold after the classification to guarantee a precision of 98% with the cost of a recall of 58%. **Results of triple validation and classification proved the benefits of mixing techniques to ensure the integrity and consistency of the knowledge graph.** Moreover, we paved the way for future investigations in the area of triple validation. With these two last chapters we automated the detection of world changes and completed the knowledge graph population system from news streams.

To sum up, the results of this thesis demonstrated how our approach can be used to build a dynamic knowledge graph that is constantly updated with the latest information from the news. The work developed in this thesis falls in the intersection of many fields, specially Natural Language Processing (NLP) and semantic technologies, but we also introduced holistic systems that benefit from image and speech recognition models. We found an opportunity in the confluence between these fields to generate a novel online knowledge graph population system.

As an industrial PhD, the efforts of this work have been focused on creating novel models that solve problems from media producers. Real world applicability of the tools created during this dissertation has been indubitably proved with the on production deployment of all the technologies developed, which generated commercial products for Vilynx. The outcomes of this industrial work have been submitted for patenting in four patent applications in the United States. However, contributions have not only been industrial, as

all the major achievements of this dissertation have been published in main conferences, producing six peer-reviewed papers. Overall, this thesis makes a significant contribution to the field of knowledge graph construction and population, and has the potential to enable a wide range of applications in various domains.

Future research on knowledge graphs should keep enlarging the confluence of techniques from different areas, with the common objective of maximising the knowledge that can be distilled from diverse sources at large scale. In this thesis we addressed some knowledge graphs general challenges, like dynamicity, self-learning, and quality assessment. We are happy to have contributed with advances on the field and believe that research should keep growing on these directions for the complete adoption of knowledge graphs, which can foster scientific progress in broad aspects of society.

Future Work

In this final section, we put into perspective the results and insights gained in each part of the thesis, and present the next steps and exploring possibilities.

Part I studied IE methods to transform unstructured data to structured knowledge. The presented method consists on an Entity Linking (EL) system, followed by a Relation Extraction (RE) model. Future work on IE should approach the joint extraction of entities and relations, with an holistic point of view. Current literature already includes many works that prove joint EL and RE provides better accuracy on both tasks [224, 222, 217]. However, non of these systems have included yet multimodal information, which has great potential, as demonstrated in this thesis.

Going down to each chapter, there is also room for future research in the individual models presented. In Chapter 2 we proposed an holistic EL model and proved its potential to enhance entity extraction from text. Future work on the EL model should study the extraction of novel entities, which still have low popularity score and introduce entity trendiness metrics. Also, there are research opportunities on the contextual models, like knowledge graph embeddings oriented to EL, and the integration of new entities into embeddings models without retraining the whole system. Finally, experiments and model training of this part of the thesis were limited by the little amount of annotated data and the difficulty to find good holistic examples. Current data used for the training and experimentation was self annotated by Vilynx's, being costly and highly time consuming to obtain. With larger datasets available, there is room for improving Named Entity Disambiguation (NED) with state-of-the art neural models. Chapter 3 focused on the second part of the IE system, the RE module. As future work, we would like to integrate knowledge graph information into the language model, which we believe would provide better context through semantic information, as some works in the literature have already proven [149].

Part II of the dissertation studied the detection and representation of events from news articles. Chapter 5 focused on event detection problems. The system was tuned in order to prior precision over recall, but in the future we want to improve the amount of false negative article assignations. In Chapter 6 we worked on the second part of VLX-Stories, which is event representation. We found difficulties on the performance evaluation, because there is no dataset that we could use to test the end-to-end system. We tested different parts of the system independently but we would like to get an objective metric of the whole performance. However, to get that we would require a human evaluation to analyze if we are missing or merging events. In the future we would like to design quality evaluation experiments to get insights on the main parts to improve.

Finally, in Part III, we studied the automatic population of a knowledge graph with novel

entities and facts, and the validation of triples to provide high quality data. Chapter 8 introduces an extension of the EL pipeline to provide emerging entity detection. So far, novel entities created are only of type '*person*', but in the future we want to extend it to other types of entities like '*locations*', '*organizations*' and '*events*'. Moreover, we want to automatize the detection of alias from aspirants by using string match and contextual features. Chapter 9 completed the knowledge graph population by validating triples inferred by the RE model, before ingesting them into the knowledge graph. Future work on this model should focus on adding contextual information thought the use of knowledge graph embeddings. Also, the models used for classification were limited by the amount of annotated data. With a larger corpus of triples we would be able to apply neural models and improve triple classification results.

To conclude, we are happy to have contributed on some of the main research lines of knowledge graph population and we encourage future researchers to keep exploring the multiple opportunities behind learning from news streams and multimodal sources.

Bibliography

[1] Sami Abu-El-Haija, Nisarg Kothari, Joonseok Lee, Paul Natsev, George Toderici, Balakrishnan Varadarajan, and Sudheendra Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016. 37, 38, 40, 106

[2] Omar Adjali, Romaric Besançon, Olivier Ferret, Hervé Le Borgne, and Brigitte Grau. Multimodal entity linking for tweets. *Advances in Information Retrieval*, 12035:463, 2020. 17, 20, 27

[3] Charu C Aggarwal and ChengXiang Zhai. A survey of text clustering algorithms. In *Mining text data*, pages 77–128. Springer, 2012. 62

[4] Eugene Agichtein and Luis Gravano. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94, 2000. 47

[5] Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C Lawrence Zitnick, Dhruv Batra, and Devi Parikh. Vqa: visual question answering. arxiv. *arXiv preprint arXiv:1505.00468*, 10, 2015. 17

[6] R Srikant Agrawal and P Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Data Bases*, pages 487–499, 1994. 70

[7] Rakesh Agrawal, Tomasz Imieliński, and Arun Swami. Mining association rules between sets of items in large databases. In *Acm sigmod record*, volume 22, pages 207–216. ACM, 1993. 70

[8] Amr Ahmed and Eric Xing. Dynamic non-parametric mixture models and the recurrent chinese restaurant process: with applications to evolutionary clustering. In *Proceedings of the 2008 SIAM International Conference on Data Mining*, pages 219–230. SIAM, 2008. 62

[9] Enrique Alfonseca and Suresh Manandhar. An unsupervised method for general named entity recognition and automated concept discovery. In *Proceedings of the 1st international conference on general WordNet, Mysore, India*, pages 34–43, 2002. 22

[10] James Allan. *Topic detection and tracking: event-based information organization*, volume 12. Springer Science & Business Media, 2012. 61

[11] Tim Althoff, Xin Luna Dong, Kevin Murphy, Safa Alai, Van Dang, and Wei Zhang. Timemachine: Timeline generation for knowledge-base entities. In *Proceedings of*

the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pages 19–28, 2015. 67

[12] Sophia Ananiadou and John Mcnaught. *Text mining for biology and biomedicine*. Citeseer, 2006. 21

[13] Approach To Real Life Decisions Are. Possibility for decision a possibilistic approach to real life decisions. 7

[14] Ron Artstein and Massimo Poesio. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34(4):555–596, 2008. 42

[15] Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. Dbpedia: A nucleus for a web of open data. *The semantic web*, pages 722–735, 2007. 1, 58

[16] Mehmet Aydar, Ozge Bozal, and Furkan Ozbay. Neural relation extraction: a survey. *arXiv e-prints*, pages arXiv–2007, 2020. 49

[17] Bogdan Babych and Anthony Hartley. Improving machine translation quality with automatic named entity recognition. In *Proceedings of the 7th International EAMT workshop on MT and other language technology tools, Improving MT through other language technology tools, Resource and tools for building MT at EACL 2003*, 2003. 21

[18] Krisztian Balog. Entity linking. In *Entity-Oriented Search*, pages 147–188. Springer, 2018. 25

[19] Krisztian Balog, Pavel Serdyukov, and Arjen P de Vries. Overview of the trec 2011 entity track. In *TREC*, volume 2011, page 11, 2011. 21

[20] Debayan Banerjee, Debanjan Chaudhuri, Mohnish Dubey, and Jens Lehmann. Pnel: Pointer network based end-to-end entity linking over knowledge graphs. In *International Semantic Web Conference*, pages 21–38. Springer, 2020. 32

[21] Roger S. Barga and Hillary Caituiro-Monge. Event correlation and pattern detection in cedr. In *EDBT Workshops*, 2006. 68

[22] Luisa Bentivogli, Pamela Forner, Claudio Giuliano, Alessandro Marchetti, Emanuele Pianta, and Kateryna Tymoshenko. Extending english ace 2005 corpus annotation with ground-truth links to wikipedia. In *Proceedings of the 2nd Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources*, pages 19–27, 2010. 20

[23] David M Blei and John D Lafferty. Dynamic topic models. In *Proceedings of the 23rd international conference on Machine learning*, pages 113–120. ACM, 2006. 62

[24] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003. 62

[25] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. AcM, 2008. 1, 51, 82

[26] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. *Advances in neural information processing systems*, 26, 2013. 97

[27] Sergey Brin. Extracting patterns and relations from the world wide web. In *International Workshop on The World Wide Web and Databases*, pages 172–183. Springer, 1998. 47

[28] Razvan Bunescu and Marius Pasca. Using encyclopedic knowledge for named entity disambiguation. 2006. 25, 26

[29] Razvan C Bunescu and Raymond J Mooney. A shortest path dependency kernel for relation extraction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 724–731. Association for Computational Linguistics, 2005. 47

[30] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka, and Tom M Mitchell. Toward an architecture for never-ending language learning. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010. 93

[31] Yee Seng Chan and Dan Roth. Exploiting background knowledge for relation extraction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 152–160. Association for Computational Linguistics, 2010. 49

[32] Hongjie Chen, Lei Xie, Cheung-Chi Leung, Xiaoming Lu, Bin Ma, and Haizhou Li. Modeling latent topics and temporal distance for story segmentation of broadcast news. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(1):112–123, 2017. 62

[33] Xinlei Chen, Abhinav Shrivastava, and Abhinav Gupta. Neil: Extracting visual knowledge from web data. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1409–1416. IEEE, 2013. 2

[34] Michael Collins. Course notes for coms w4705: Language modeling, 2011. 48

[35] Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999. 22

[36] Kevyn Collins-Thompson and booktitle=Proceedings of the 14th ACM international conference on Information and knowledge management pages=704–711 year=2005 organization=ACM Callan, Jamie. Query expansion using random walk models. 62

[37] Jack G Conrad and Michael Bender. Semi-supervised events clustering in news retrieval. In *NewsIR@ ECIR*, pages 21–26, 2016. 57, 62

[38] Andras Csomai and Rada Mihalcea. Linking documents to encyclopedic knowledge. *IEEE Intelligent Systems*, 23(5):34–41, 2008. 25, 26

[39] Silviu Cucerzan. Large-scale named entity disambiguation based on wikipedia data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007. 20, 26

[40] Aron Culotta and Jeffrey Sorensen. Dependency tree kernels for relation extraction. In *Proceedings of the 42nd annual meeting on association for computational linguistics*, page 423. Association for Computational Linguistics, 2004. 47

[41] Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019. 48

[42] Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval. *arXiv preprint arXiv:2010.00904*, 2020. 26

[43] Gianluca Demartini, Tereza Iofciu, and Arjen P De Vries. Overview of the inex 2009 entity ranking track. In *International Workshop of the Initiative for the Evaluation of XML Retrieval*, pages 254–264. Springer, 2009. 21

[44] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 26, 48, 49, 50

[45] Dua Dheeru and Efi Karra Taniskidou. UCI machine learning repository, 2017. 65, 66, 107

[46] Bayu Distiawan, Gerhard Weikum, Jianzhong Qi, and Rui Zhang. Neural relation extraction for knowledge base enrichment. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 229–240, 2019. 97

[47] George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie M Strassel, and Ralph M Weischedel. The automatic content extraction (ace) program-tasks, data, and evaluation. In *Lrec*, volume 2, pages 837–840. Lisbon, 2004. 21

[48] Tiansi Dong, Zhigang Wang, Juanzi Li, Christian Bauckhage, and Armin B Cremers. Triple classification using regions and fine-grained entity typing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 77–85, 2019. 97

[49] Xin Dong, Evgeniy Gabrilovich, Geremy Heitz, Wilko Horn, Ni Lao, Kevin Murphy, Thomas Strohmann, Shaohua Sun, and Wei Zhang. Knowledge vault: A web-scale approach to probabilistic knowledge fusion. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 601–610, 2014. 10, 79, 80

[50] Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 277–285, 2010. 22

[51] Fredo Erxleben, Michael Günther, Markus Krötzsch, Julian Mendez, and Denny Vrandečić. Introducing wikidata to the linked data web. In *International semantic web conference*, pages 50–65. Springer, 2014. 11

[52] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Kdd*, volume 96, pages 226–231, 1996. 62, 64, 71

[53] Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74, 2008. 16

[54] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134, 2005. 22, 47

[55] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1535–1545. Association for Computational Linguistics, 2011. 80

[56] Michael Färber, Achim Rettinger, and Boulos El Asmar. On emerging entity detection. In *European Knowledge Acquisition Workshop*, pages 223–238. Springer, 2016. 86, 87

[57] Dimitra Farmakiotou, Vangelis Karkaletsis, John Koutsias, George Sigletos, Constantine D Spyropoulos, and Panagiotis Stamatopoulos. Rule-based named entity recognition for greek financial texts. In *Proceedings of the Workshop on Computational lexicography and Multimedia Dictionaries (COMLEX 2000)*, pages 75–78, 2000. 21

[58] Christiane Fellbaum. *WordNet*. Wiley Online Library, 1998. 11

[59] Delia Fernández, David Varas, Joan Espadaler, Issey Masuda, Jordi Ferreira, Alejandro Woodward, David Rodríguez, Xavier Giró Nieto, Juan Carlos Riveiro, and Elisenda Bou Balust. Vits: Video tagging system from massive web multimedia collections. In *Proceedings of the 5th Workshop on Web-scale Vision and Social Media (VSM)*, pages 337–346. IEEE Press, 2017. 51, 82

[60] Dèlia Fernàndez-Cañellas, Joan Espadaler, David Rodriguez, Blai Garolera, Gemma Canet, Aleix Colom, Joan Marco Rimmek, Xavier Giro-i Nieto, Elisenda Bou, and Juan Carlos Riveiro. Vlx-stories: Building an online event knowledge base with emerging entity detection. In *International Semantic Web Conference*, pages 382–399. Springer, 2019. 82

[61] Dèlia Fernàndez-Cañellas, Joan Marco Rimmek, Joan Espadaler, Blai Garolera, Adrià Barja, Marc Codina, Marc Sastre, Xavier Giro-i Nieto, Juan Carlos Riveiro, and Elisenda Bou-Balust. Enhancing online knowledge graph population with semantic knowledge. In *International Semantic Web Conference*, pages 183–200. Springer, 2020. 51, 107

[62] Paolo Ferragina and Ugo Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1625–1628, 2010. 25, 26

[63] D. A. Ferrucci. Introduction to "this is watson". *IBM Journal of Research and Development*, 56(3.4):1:1–1:15, 2012. 15

[64] David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A Kalyanpur, Adam Lally, J William Murdock, Eric Nyberg, John Prager, et al. Building watson: An overview of the deepqa project. *AI magazine*, 31(3):59–79, 2010. 1

[65] Jonathan G Fiscus and George R Doddington. Topic detection and tracking evaluation overview. In *Topic detection and tracking*, pages 17–31. Springer, 2002. 17

[66] Octavian-Eugen Ganea and Thomas Hofmann. Deep joint entity disambiguation with local neural attention. *arXiv preprint arXiv:1704.04920*, 2017. 26, 27

[67] Abhishek Gattani, Digvijay S Lamba, Nikesh Garera, Mitul Tiwari, Xiaoyong Chai, Sanjib Das, Sri Subramaniam, Anand Rajaraman, Venky Harinarayan, and AnHai Doan. Entity extraction, linking, classification, and tagging for social media: a wikipedia-based approach. *Proceedings of the VLDB Endowment*, 6(11):1126–1137, 2013. 25

[68] Jose Emilio Labra Gayo, Eric Prud'hommeaux, Harold R Solbrig, and Jose María Álvarez Rodríguez. Validating and describing linked data portals using rdf shape expressions. In *LDQ@ SEMANTICS*, 2014. 96

[69] Amir Globerson, Nevena Lazic, Soumen Chakrabarti, Amarnag Subramanya, Michael Ringaard, and Fernando Pereira. Collective entity resolution with multi-focal attention. 2016. 26

[70] Dihong Gong, Daisy Zhe Wang, and Yang Peng. Multimodal learning for web information extraction. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 288–296, 2017. 17

[71] Simon Gottschalk and Elena Demidova. Eventkg: A multilingual event-centric temporal knowledge graph. In *European Semantic Web Conference*, pages 272–287. Springer, 2018. 68

[72] Simon Gottschalk, Elena Demidova, Viola Bernacchi, and Richard Rogers. Ongoing events in wikipedia: a cross-lingual case study. In *Proceedings of the 2017 ACM on Web Science Conference*, pages 387–388, 2017. 67

[73] Ralph Grishman and Beth M Sundheim. Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996. 19, 21

[74] Thomas R Gruber. A translation approach to portable ontology specifications. *Knowledge acquisition*, 5(2):199–220, 1993. 7

[75] Xifeng Guo, Long Gao, Xinwang Liu, and Jianping Yin. Improved deep embedded clustering with local structure preservation. In *IJCAI*, pages 1753–1759, 2017. 57

[76] Zhaochen Guo and Denilson Barbosa. Robust named entity disambiguation with random walks. *Semantic Web*, 9(4):459–479, 2018. 26

[77] Zhou GuoDong, Su Jian, Zhang Jie, and Zhang Min. Exploring various knowledge in relation extraction. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 427–434. Association for Computational Linguistics, 2005. 47

[78] Felix Hamborg, Corinna Breitinger, Moritz Schubotz, Soeren Lachnit, and Bela Gipp. Extraction of main event descriptors from news articles by answering the journalistic five w and one h questions. In *JCDL*, pages 339–340, 2018. 57

[79] Felix Hamborg, Soeren Lachnit, Moritz Schubotz, Thomas Hepp, and Bela Gipp. Giveme5w: main event retrieval from news articles by extraction of the five journalistic w questions. In *International Conference on Information*, pages 356–366. Springer, 2018. 57

[80] Xianpei Han, Le Sun, and Jun Zhao. Collective entity linking in web text: a graph-based method. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, pages 765–774, 2011. 26

[81] Xu Han, Tianyu Gao, Yuan Yao, Deming Ye, Zhiyuan Liu, and Maosong Sun. OpenNRE: An open and extensible toolkit for neural relation extraction. In *Proceedings of EMNLP-IJCNLP: System Demonstrations*, pages 169–174, 2019. 45, 79

[82] Xu Han, Hao Zhu, Pengfei Yu, Ziyun Wang, Yuan Yao, Zhiyuan Liu, and Maosong Sun. Fewrel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation. *arXiv preprint arXiv:1810.10147*, 2018. 51, 52

[83] Leonhard Hennig, Danuta Ploch, Daniel Prawdzik, Benjamin Armbruster, H Düwiger, Ernesto William De Luca, and S Albayrak. Spiga-a multilingual news aggregator. *Proceedings of GSCL*, 2011, 2011. 57, 59, 62, 63, 64, 68, 70

[84] Rob High. The era of cognitive systems: An inside look at ibm watson and how it works. *IBM Corporation, Redbooks*, pages 1–16, 2012. 15

[85] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 65

[86] Johannes Hoffart, Yasemin Altun, and Gerhard Weikum. Discovering emerging entities with ambiguous names. In *Proceedings of the 23rd international conference on World wide web*, pages 385–396. ACM, 2014. 86, 87

[87] Johannes Hoffart, Dragan Milchevski, Gerhard Weikum, Avishek Anand, and Jaspreet Singh. The knowledge awakens: Keeping knowledge bases fresh with emerging entities. In *Proceedings of the 25th International Conference Companion on World Wide Web*, pages 203–206. International World Wide Web Conferences Steering Committee, 2016. 2, 79, 87

[88] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792. Association for Computational Linguistics, 2011. 20, 26

[89] Konrad Höffner, Sebastian Walter, Edgard Marx, Ricardo Usbeck, Jens Lehmann, and Axel-Cyrille Ngonga Ngomo. Survey on challenges of question answering in the semantic web. *Semantic Web*, 8(6):895–920, 2017. 67

[90] Thomas Hofmann. Probabilistic latent semantic analysis. In *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pages 289–296. Morgan Kaufmann Publishers Inc., 1999. 62

[91] Aidan Hogan, Andreas Harth, Alexandre Passant, Stefan Decker, and Axel Polleres. Weaving the pedantic web. 2010. 96

[92] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020. 30

[93] Wen Hua, Kai Zheng, and Xiaofang Zhou. Microblog entity linking with social temporal context. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, pages 1761–1775, 2015. 27

[94] Heng Ji and Ralph Grishman. Data selection in semi-supervised learning for name tagging. In *Proceedings of the Workshop on Information Extraction Beyond The Document*, pages 48–55, 2006. 22

[95] Heng Ji and Ralph Grishman. Refining event extraction through cross-document inference. *Proceedings of ACL-08: HLT*, pages 254–262, 2008. 57

[96] Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. Overview of the tac 2010 knowledge base population track. In *Third text analysis conference (TAC 2010)*, volume 3, pages 3–3, 2010. 20, 86

[97] Heng Ji, Xiaoman Pan, Boliang Zhang, Joel Nothman, James Mayfield, Paul McNamee, Cash Costello, and Sydney Informatics Hub. Overview of tac-kbp2017 13 languages entity discovery and linking. In *TAC*, 2017. 79

[98] Jing Jiang and ChengXiang Zhai. A systematic exploration of the feature space for relation extraction. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 113–120, 2007. 47

[99] Xiaotian Jiang, Quan Wang, Peng Li, and Bin Wang. Relation extraction with multi-instance multi-label convolutional neural networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1471–1480, 2016. 48

[100] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020. 49, 52

[101] Brendan Jou, Hongzhi Li, Joseph G Ellis, Daniel Morozoff-Abegauz, and Shih-Fu Chang. Structured exploration of who, what, when, and where in heterogeneous multimedia news sources. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 357–360. ACM, 2013. 72

[102] Nanda Kambhatla. Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive poster and demonstration sessions*, pages 22–es, 2004. 47

[103] Rijula Kar, Susmija Reddy, Sourangshu Bhattacharya, Anirban Dasgupta, and Soumen Chakrabarti. Task-specific representation learning for web-scale entity disambiguation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018. 24

[104] Mayank Kejriwal. *Domain-Specific Knowledge Graph Construction*. Springer, 2019. 7

[105] Holger Knublauch and Dimitris Kontokostas. Shapes constraint language (shacl). *W3C Candidate Recommendation*, 11(8), 2017. 96

[106] Georgi Kobilarov, Tom Scott, Yves Raimond, Silver Oliver, Chris Sizemore, Michael Smethurst, Christian Bizer, and Robert Lee. Media meets semantic web–how the bbc uses dbpedia and linked data to make connections. In *European Semantic Web Conference*, pages 723–737. Springer, 2009. 10

[107] Dimitris Kontokostas, Amrapali Zaveri, Sören Auer, and Jens Lehmann. Triplecheckmate: A tool for crowdsourcing the quality assessment of linked data. In *International Conference on Knowledge Engineering and the Semantic Web*, pages 265–272. Springer, 2013. 95

[108] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 457–466, 2009. 25, 26

[109] Shantanu Kumar. A survey of deep learning methods for relation extraction. *arXiv preprint arXiv:1705.03645*, 2017. 47, 48

[110] Haewoon Kwak and Jisun An. A first look at global news coverage of disasters by using the gdelt dataset. In *International Conference on Social Informatics*, pages 300–308. Springer, 2014. 58

[111] Carl Lagoze and Jane Hunter. The abc ontology and model. In *International Conference on Dublin Core and Metadata Applications*, pages 160–176, 2001. 69

[112] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016. 49

[113] Phong Le and Ivan Titov. Improving entity linking by modeling latent relations between mentions. *arXiv preprint arXiv:1804.10637*, 2018. 20, 26

[114] Gregor Leban, Blaz Fortuna, and Marko Grobelnik. Using news articles for real-time cross-lingual event detection and filtering. In *NewsIR@ ECIR*, pages 33–38, 2016. 57, 68, 70

[115] Kalev Leetaru and Philip A Schrodt. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer, 2013. 58

[116] Douglas B Lenat. Cyc: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11):33–38, 1995. 1

[117] Hongzhi Li, Joseph G Ellis, Heng Ji, and Shih-Fu Chang. Event specific multimodal pattern mining for knowledge base construction. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 821–830. ACM, 2016. 68, 70

[118] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition. *IEEE Transactions on Knowledge and Data Engineering*, 2020. 22

[119] Qi Li, Heng Ji, and Liang Huang. Joint event extraction via structured prediction with global features. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 73–82, 2013. 68

[120] Weixin Li, Jungseock Joo, Hang Qi, and Song-Chun Zhu. Joint image-text news topic detection and tracking by multimodal topic and-or graph. *IEEE Transactions on Multimedia*, 19(2):367–381, 2017. 62

[121] Thomas Lin, Oren Etzioni, et al. Identifying functional relations in web text. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 1266–1276. Association for Computational Linguistics, 2010. 17

[122] Xueling Lin, Haoyang Li, Hao Xin, Zijian Li, and Lei Chen. Kbpearl: a knowledge base population system supported by joint entity and relation linking. *Proceedings of the VLDB Endowment*, 13(7):1035–1049, 2020. 80

[123] Yankai Lin, Shiqi Shen, Zhiyuan Liu, Huanbo Luan, and Maosong Sun. Neural relation extraction with selective attention over instances. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2124–2133, 2016. 48

[124] ChunYang Liu, WenBo Sun, WenHan Chao, and Wanxiang Che. Convolution neural network for relation extraction. In *International Conference on Advanced Data Mining and Applications*, pages 231–242. Springer, 2013. 48

[125] Gang Luo, Xiaojiang Huang, Chin-Yew Lin, and Zaiqing Nie. Joint entity recognition and disambiguation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 879–888, 2015. 20, 27

[126] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*, pages 55–60, 2014. 58

[127] Jose L Martinez-Rodriguez, Aidan Hogan, and Ivan Lopez-Arevalo. Information extraction meets the semantic web: A survey. *Semantic Web*, (Preprint):1–81, 2018. 2, 79

[128] Kathleen R McKeown, Regina Barzilay, David Evans, Vasileios Hatzivassiloglou, Judith L Klavans, Ani Nenkova, Carl Sable, Barry Schiffman, and Sergey Sigelman. Tracking and summarizing news on a daily basis with columbia's newsblaster. In *Proceedings of the second international conference on Human Language Technology Research*, pages 280–285. Morgan Kaufmann Publishers Inc., 2002. 59

[129] Filipe Mesquita, Matteo Cannaviccio, Jordan Schmidek, Paramita Mirza, and Denilson Barbosa. Knowledgenet: A benchmark dataset for knowledge base population. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 749–758, 2019. 51, 52

[130] George A Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995. 22, 58

[131] David Milne and Ian H Witten. Learning to link with wikipedia. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 509–518, 2008. 25, 26, 33

[132] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 1003–1011. Association for Computational Linguistics, 2009. 46, 48

[133] Tom Mitchell, William Cohen, Estevam Hruschka, Partha Talukdar, Bo Yang, Justin Betteridge, Andrew Carlson, B Dalvi, Matt Gardner, Bryan Kisiel, et al. Never-ending learning. *Communications of the ACM*, 61(5):103–115, 2018. 79, 80

[134] Diego Mollá, Menno Van Zaanen, Daniel Smith, et al. Named entity recognition for question answering. 2006. 21

[135] Seungwhan Moon, Leonardo Neves, and Vitor Carvalho. Multimodal named entity disambiguation for noisy social media posts. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2000–2008, 2018. 20, 27

[136] Raymond J Mooney and Razvan C Bunescu. Subsequence kernels for relation extraction. In *Advances in neural information processing systems*, pages 171–178, 2006. 47

[137] David Nadeau. *Semi-supervised named entity recognition: learning to recognize 100 entity types with little supervision*. PhD thesis, University of Ottawa, 2007. 22

[138] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26, 2007. 22, 49

[139] David Nadeau, Peter D Turney, and Stan Matwin. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In *Conference of the Canadian society for computational studies of intelligence*, pages 266–277. Springer, 2006. 22

[140] Kok Wah Ng, Flora S Tsai, Lihui Chen, and Kiat Chong Goh. Novelty detection for text documents using named entity recognition. In *2007 6th international conference on information, communications & signal processing*, pages 1–5. IEEE, 2007. 63

[141] Thien Huu Nguyen and Ralph Grishman. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48, 2015. 48

[142] Natalya F Noy, Deborah L McGuinness, et al. Ontology development 101: A guide to creating your first ontology, 2001. 7

[143] Italo L Oliveira, Renato Fileto, René Speck, Luís PF Garcia, Diego Moussallem, and Jens Lehmann. Towards holistic entity linking: Survey and directions. *Information Systems*, 95:101624, 2021. 27

[144] Gabriele Paolacci, Jesse Chandler, and Panagiotis G Ipeirotis. Running experiments on amazon mechanical turk. 2010. 37

[145] Rebecca Passonneau, Nizar Habash, and Owen Rambow. Inter-annotator agreement on a multilingual semantic annotation task. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC)*, pages 1951–1956, 2006. 42

[146] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 101

[147] Thomas Pellissier Tanon, Denny Vrandečić, Sebastian Schaffert, Thomas Steiner, and Lydia Pintscher. From freebase to wikidata: The great migration. In *Proceedings of the 25th International Conference on World Wide Web*, pages 1419–1428. International World Wide Web Conferences Steering Committee, 2016. 10

[148] Maria Pershina, Yifan He, and Ralph Grishman. Personalized page rank for named entity disambiguation. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 238–243, 2015. 26

[149] Matthew E. Peters, Mark Neumann, IV RobertLLogan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. Knowledge enhanced contextual word representations. In *EMNLP/IJCNLP*, 2019. 49, 53, 111

[150] Axel Polleres, Aidan Hogan, Andreas Harth, and Stefan Decker. Can we ever catch up with the web? *Semantic Web*, 1(1, 2):45–52, 2010. 57

[151] Jay M Ponte and W Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–281. ACM, 1998. 62

[152] Bruno Pouliquen, Ralf Steinberger, and Olivier Deguernel. Story tracking: linking similar news over time and across languages. In *Proceedings of the workshop on Multi-source Multilingual Information Extraction and Summarization*, pages 49–56. Association for Computational Linguistics, 2008. 59

[153] Radityo Eko Prasojo, Mouna Kacimi, and Werner Nutt. Modeling and summarizing news events using semantic triples. In *European Semantic Web Conference*, pages 512–527. Springer, 2018. 68

[154] Pengda Qin, Weiran Xu, and William Yang Wang. Robust distant supervision relation extraction via deep reinforcement learning. *arXiv preprint arXiv:1805.09927*, 2018. 48

[155] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Blog*, 1(8):9, 2019. 48

[156] Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and global algorithms for disambiguation to wikipedia. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 1375–1384, 2011. 25

[157] Xiang Ren, Zeqiu Wu, Wenqi He, Meng Qu, Clare R Voss, Heng Ji, Tarek F Abdelzaher, and Jiawei Han. Cotype: Joint extraction of typed entities and relations with knowledge bases. In *Proceedings of the 26th International Conference on World Wide Web*, pages 1015–1024, 2017. 48

[158] Cecilia Reyes-Pena and Mireya Tovar-Vidal. Ontology: Components and evaluation, a review. *Research in Computing Science*, 148:257–265, 2019. 8

[159] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling relations and their mentions without labeled text. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 148–163. Springer, 2010. 46, 48

[160] Ellen Riloff, Rosie Jones, et al. Learning dictionaries for information extraction by multi-level bootstrapping. In *AAAI/IAAI*, pages 474–479, 1999. 22

[161] Alan Ritter, Oren Etzioni, et al. A latent dirichlet allocation method for selectional preferences. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 424–434. Association for Computational Linguistics, 2010. 17

[162] Benjamin Rosenfeld and Ronen Feldman. Ures: an unsupervised web relation extraction system. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 667–674. Association for Computational Linguistics, 2006. 47

[163] Marco Rospocher, Marieke van Erp, Piek Vossen, Antske Fokkens, Itziar Aldabe, German Rigau, Aitor Soroa, Thomas Ploeger, and Tessel Bogaard. Building event-centric knowledge graphs from news. *Journal of Web Semantics*, 37:132–151, 2016. 57, 68

[164] Dan Roth and Wen-tau Yih. Global inference for entity and relation identification via a linear programming formulation. *Introduction to statistical relational learning*, pages 553–580, 2007. 49

[165] Jan Rupnik, Andrej Muhic, Gregor Leban, Primoz Skraba, Blaz Fortuna, and Marko Grobelnik. News across languages-cross-lingual document similarity and event tracking. *Journal of Artificial Intelligence Research*, 55:283–316, 2016. 59

[166] Arthur G Ryman, Arnaud Le Hors, and Steve Speicher. Oslc resource shape: A language for defining constraints on linked data. *LDOW*, 996, 2013. 96

[167] Horacio Saggion, Adam Funk, Diana Maynard, and Kalina Bontcheva. Ontology-based information extraction for business intelligence. In *The Semantic Web*, pages 843–856. Springer, 2007. 21

[168] Tomer Sagi, Yael Wolf, and Katja Hose. How new is the (rdf) news? In *Companion Proceedings of The 2019 World Wide Web Conference*, pages 714–721. ACM, 2019. 2, 79

[169] Gerard Salton. Automatic text processing: The transformation, analysis, and retrieval of. *Reading: Addison-Wesley*, 1989. 62

[170] Evan Sandhaus. semantic technology at the new york times: lessons learned and future directions. In *International Semantic Web Conference*, pages 355–355. Springer, 2010. 10

[171] Erik F Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*, 2003. 21

[172] Sunita Sarawagi et al. Information extraction. *Foundations and Trends® in Databases*, 1(3):261–377, 2008. 15, 80

[173] Michael Schmitz, Robert Bart, Stephen Soderland, Oren Etzioni, et al. Open language learning for information extraction. In *Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning*, pages 523–534. Association for Computational Linguistics, 2012. 93

[174] Philip A Schrodt, John Beieler, and Muhammed Idris. Three'sa charm?: open event data coding with el: Diablo, petrarch, and the open event data alliance. In *ISA Annual Convention*, 2014. 58

[175] Ozge Sevgili, Artem Shelmanov, Mikhail Arkhipov, Alexander Panchenko, and Chris Biemann. Neural entity linking: A survey of models based on deep learning. *arXiv preprint arXiv:2006.00575*, 2020. 24, 26

[176] Chirag Shah, W Bruce Croft, and David Jensen. Representing documents with named entities for story link detection (sld). In *Proceedings of the 15th ACM international conference on Information and knowledge management*, pages 868–869. ACM, 2006. 62, 63

[177] Wei Shen, Jianyong Wang, and Jiawei Han. Entity linking with a knowledge base: Issues, techniques, and solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460, 2014. 24, 79

[178] Peng Shi and Jimmy Lin. Simple bert models for relation extraction and semantic role labeling. *CoRR*, abs/1904.05255, 2019. 49

[179] Yusuke Shinyama and Satoshi Sekine. Named entity discovery using comparable news articles. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 848–853, 2004. 22

[180] Nakatani Shuyo. Language detection library for java, 2010. 40

[181] Avirup Sil and Alexander Yates. Re-ranking for joint named-entity recognition and linking. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 2369–2374. ACM, 2013. 27

[182] Jane B Singer. Five ws and an h: Digital challenges in newspaper newsrooms and boardrooms. *The International Journal on Media Management*, 2008. 57, 60

[183] Livio Baldini Soares, Nicholas FitzGerald, Jeffrey Ling, and Tom Kwiatkowski. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2895–2905, 2019. vii, 49, 50, 52, 53, 106

[184] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. Reasoning with neural tensor networks for knowledge base completion. In *Advances in neural information processing systems*, pages 926–934, 2013. 97

[185] Blerina Spahiu, Andrea Maurino, and Matteo Palmonari. Towards improving the quality of knowledge graphs with data-driven ontology patterns and shacl. In *ISWC (Best Workshop Papers)*, pages 103–117, 2018. 96

[186] Michael Steinbach, George Karypis, Vipin Kumar, et al. A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pages 525–526. Boston, 2000. 62

[187] Josef Steinberger. Mediagist: A cross-lingual analyser of aggregated news and commentaries. *Proceedings of ACL-2016 System Demonstrations*, pages 145–150, 2016. 57

[188] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. Yago: a core of semantic knowledge. In *Proceedings of the 16th international conference on World Wide Web*, pages 697–706. ACM, 2007. 1

[189] Philippe Thomas, Johannes Kirschnick, Leonhard Hennig, Renlong Ai, Sven Schmeier, Holmer Hemsen, Feiyu Xu, and Hans Uszkoreit. Streaming text analytics for real-time event recognition. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 750–757, 2017. 59

[190] Dominik Tomaszuk. Rdf validation: A brief survey. In *International Conference: Beyond Databases, Architectures and Structures*, pages 344–355. Springer, 2017. 96

[191] Gerald Töpper, Magnus Knuth, and Harald Sack. Dbpedia ontology enrichment for inconsistency detection. In *Proceedings of the 8th International Conference on Semantic Systems*, pages 33–40, 2012. 96

[192] Tuan Tran, Nam Khanh Tran, Teka Hadgu Asmelash, and Robert Jäschke. Semantic annotation for microblog topics using wikipedia temporal information. *arXiv preprint arXiv:1701.03939*, 2017. 27

[193] Johan Ugander, Brian Karrer, Lars Backstrom, and Cameron Marlow. The anatomy of the facebook social graph. *arXiv preprint arXiv:1111.4503*, 2011. 1

[194] Willem Robert Van Hage, Véronique Malaisé, Roxane Segers, Laura Hollink, and Guus Schreiber. Design and use of the simple event model (sem). *Web Semantics: Science, Services and Agents on the World Wide Web*, 9(2):128–136, 2011. 58, 69

[195] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 48, 49

[196] Piek Vossen, Rodrigo Agerri, Itziar Aldabe, Agata Cybulska, Marieke van Erp, Antske Fokkens, Egoitz Laparra, Anne-Lyse Minard, Alessio Palmero Aprosio, German Rigau, et al. Newsreader: Using knowledge resources in a cross-lingual reading machine to generate more knowledge from massive streams of news. *Knowledge-Based Systems*, 110:60–85, 2016. 58, 68, 70

[197] Denny Vrandečić and Markus Krötzsch. Wikidata: a free collaborative knowledgebase. *Communications of the ACM*, 57(10):78–85, 2014. 1, 82

[198] Wei Wang. *Event Detection and Extraction from News Articles*. PhD thesis, Virginia Tech, 2018. 58

[199] Wei Wang and Dongyan Zhao. Ontology-based event modeling for semantic understanding of chinese news story. In *Natural Language Processing and Chinese Computing*, pages 58–68. Springer, 2012. 69

[200] Zihao Wang, Xihui Liu, Hongsheng Li, Lu Sheng, Junjie Yan, Xiaogang Wang, and Jing Shao. Camp: Cross-modal adaptive message passing for text-image retrieval. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5764–5773, 2019. 17

[201] Michael D Ward, Andreas Beger, Josh Cutler, Matt Dickenson, Cassy Dorff, and Ben Radford. Comparing gdelt and icews event data. *Analysis*, 21(1):267–97, 2013. 58

[202] Mark Wick. *GeoNames*. GeoNames, 2006. 11

[203] Daya C Wimalasuriya and Dejing Dou. Ontology-based information extraction: An introduction and a survey of current approaches, 2010. 16

[204] Shanchan Wu and Yifan He. Enriching pre-trained language model with entity information for relation classification. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, pages 2361–2364, 2019. 49

[205] Zhaohui Wu, Chen Liang, and C Lee Giles. Storybase: Towards building a knowledge base for news events. *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, pages 133–138, 2015. 57, 68

[206] Zhaohui Wu, Yang Song, and C Giles. Exploring multiple feature spaces for novel entity discovery. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016. 87

[207] Ikuya Yamada, Koki Washio, Hiroyuki Shindo, and Yuji Matsumoto. Global entity disambiguation with pretrained contextualized embeddings of words and entities. *arXiv preprint arXiv:1909.00426*, 2019. 26

[208] Rong Yan and Alexander G Hauptmann. A review of text and image retrieval approaches for broadcast news video. *Information Retrieval*, 10(4-5):445–484, 2007. 17

[209] Liang Yao, Chengsheng Mao, and Yuan Luo. Kg-bert: Bert for knowledge graph completion. *arXiv preprint arXiv:1909.03193*, 2019. 49

[210] Alexander Yates, Michele Banko, Matthew Broadhead, Michael J Cafarella, Oren Etzioni, and Stephen Soderland. Textrunner: open information extraction on the web. In *Proceedings of Human Language Technologies: The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)*, pages 25–26, 2007. 47, 80

[211] Quanzeng You, Hailin Jin, Zhaowen Wang, Chen Fang, and Jiebo Luo. Image captioning with semantic attention. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4651–4659, 2016. 17

[212] Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. Kernel methods for relation extraction. *Journal of machine learning research*, 3(Feb):1083–1106, 2003. 47

[213] Daojian Zeng, Kang Liu, Yubo Chen, and Jun Zhao. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pages 1753–1762, 2015. 48

[214] Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, Jun Zhao, et al. Relation classification via convolutional deep neural network. 2014. 48

[215] Dongxu Zhang and Dong Wang. Relation classification via recurrent neural network. *arXiv preprint arXiv:1508.01006*, 2015. 51, 52

[216] Lei Zhang, Tianxing Wu, Liang Xu, Meng Wang, Guilin Qi, and Harald Sack. Emerging entity discovery using web sources. In *China Conference on Knowledge Graph and Semantic Computing*, pages 175–184. Springer, 2019. 87

[217] Sheng Zhang, Patrick Ng, Zhiguo Wang, and Bing Xiang. Reknow: Enhanced knowledge for joint entity and relation extraction. *arXiv preprint arXiv:2206.05123*, 2022. 111

[218] Tongtao Zhang, Spencer Whitehead, Hanwang Zhang, Hongzhi Li, Joseph Ellis, Lifu Huang, Wei Liu, Heng Ji, and Shih-Fu Chang. Improving event extraction via multimodal integration. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 270–278. ACM, 2017. 57, 68, 70

[219] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 35–45, 2017. 45, 51, 79

[220] Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 35–45, 2017. 106

[221] Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. ERNIE: Enhanced language representation with informative entities. In *Proceedings of ACL 2019*, 2019. 49, 52

[222] Zhenyu Zhang, Xiaobo Sind, Tingwen Liu, Zheng Fang, and Quangang Li. Joint entity linking and relation extraction with neural networks for knowledge base population. In *2020 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2020. 111

[223] Zhiyuan Zhang, Bing Wang, Faisal Ahmed, IV Ramakrishnan, Rong Zhao, Asa Viccellio, and Klaus Mueller. The five ws for information visualization with application to healthcare informatics. *IEEE transactions on visualization and computer graphics*, 19(11):1895–1910, 2013. 68

[224] Suncong Zheng, Yuexing Hao, Dongyuan Lu, Hongyun Bao, Jiaming Xu, Hongwei Hao, and Bo Xu. Joint entity and relation extraction based on a hybrid neural network. *Neurocomputing*, 257:59–66, 2017. 48, 111

[225] Yuke Zhu, Alireza Fathi, and Li Fei-Fei. Reasoning about object affordances in a knowledge base representation. In *European conference on computer vision*, pages 408–424. Springer, 2014. 2

[226] Yuke Zhu, Ce Zhang, Christopher Ré, and Li Fei-Fei. Building a large-scale multimodal knowledge base system for answering visual queries. *arXiv preprint arXiv:1507.05670*, 2015. 2