Text Frame Detector: Slot Filling Based On Domain Knowledge Bases

Martina Miliani, Lucia C. Passaro and **Alessandro Lenci**
CoLing Lab, Dipartimento di Filologia, Letteratura e Linguistica (FiLeLi), Università di Pisa
martina.miliani@fileli.unipi.it
lucia.passaro@fileli.unipi.it
alessandro.lenci@unipi.it

Abstract

English. In this paper we present a system called *Text Frame Detector* (TFD) which aims at populating a frame-based ontology in a graph-based structure. Our system organizes textual information into frames, according to a predefined set of semantically informed patterns linking pre-coded information such as named entities, simple and complex terms. Given the semi-automatic expansion of such information with word embeddings, the system can be easily adapted to new domains.

1 Introduction

Textual data are still the most widespread content around the Web (Smirnova and Cudré-Mauroux, 2018). Information Extraction (IE) is a key task to structure textual information and make it machine understandable. IE can be modelled as the process of filling *semantic frames* specified within a domain ontology and consisting of a collection of slots typed with their possible values (Minsky, 1974; Jurafsky and Martin, 2018). Therefore, each frame can be seen as a set of relations whose participants are the values of the slots. Following Jean-Louis et al. (2011), we refer to such relations as complex relations, namely any n-ary relation among typed entities.

Relation extraction techniques have been widely applied to populate semantic frames (Surdeanu, 2013; Zhenjun et al., 2017). However, both supervised and unsupervised methods have shown their limits. On the one hand, supervised approaches (Zelenko et al., 2003; Mooney and Bunescu, 2005; Nguyen and Grishman, 2015; Zhang et al., 2017) model frame filling as a classification task, hence they require labelled data, with the consequent high cost of long annotation time. On the other hand, unsupervised approaches do not need any training data, but mapping extraction results onto predefined relations or ontologies is often quite challenging with this kind of methods (Fader et al., 2011).

Moreover, semi-supervised methods exploit bootstrap learning, so that any new relation requires a small set of labelled data to be extracted (Agichtein and Gravano, 2000; Chen et al., 2006; Weld et al., 2008).

Finally, another kind of approach has been proposed, which relies on knowledge bases (KBs) to produce training data. Introduced by Mintz et al. (2009), *distant supervision* detects relations on semantically annotated texts where entities which co-occur in the same sentence match with entity-pairs contained in the KB. Then a classifier is trained using features extracted from the annotated relations (Smirnova and Cudré-Mauroux, 2018). Although this approach has been proven to be effective, the supervised step could suffer from scarce amount of data, especially if the relations occur with low frequency in small corpora.

In this paper, we present a system to populate a frame-based ontology, whose values are stored in a graph-based structure. Our method exploits some aspects of distant supervision, leveraging on domain specific KB to infer the relations, and populates the frames with specific information (i.e., the participants) as well as the portions of text (i.e., the snippets) which contain them. Thus, the output of the system for a single frame is a set of snippets, one for each of its slots. Each snippet is also associated with a weight encoding how likely it is expected to contain the information about a certain relation. Such a weight is calculated with a scoring function based on similarity measures and textual distance information. The system has been tested on the administrative domain, with the goal of gathering information related to taxes and agenda events. Indeed, since the KB can be semi-automatically enriched with Named Entities (NEs) and vocabularies of simple and complex terms, our approach can be easily adapted to different domains. Furthermore, system recall can be increased by expanding the frame and attribute vocabulary by exploiting word embeddings (Mikolov et al., 2013).

Our approach differs from existing systems like PIKES (Concoglioniti et al., 2016), Framester (Gangemi et al., 2016), FRED (Gangemi et al., 2017), and Framebase (Rouces et al., 2015) primarily for the notion of semantic frame we have adopted. The works above are mainly based on Fillmore's (1976) definition of frame as encoded in FrameNet: frames and associated roles describe situations evoked by lexical expressions (i.e. *Lexical Units*). In our system a frame represents a domain entity (e.g. "tax") by means of attributes and relations associated to that domain. Unlike FrameNet frames, these attributes and relations are activated by a set of distributed lexico-syntactic cues.

This paper is structured as follows: in section 2 we describe the general methodology of the system, we define terminology and notation and we describe the main features of the proposed approach. The system implementation is illustrated in section 3, which shows the extraction algorithm as well as the indexing methods in the knowledge graph. Evaluation and results are reported in section 4.

2 Methodology

Following Riedel et al. (2010), we assume that "if two entities $\langle e_1, e_2 \rangle$ participate in a relation $\langle r \rangle$, then there is at least one sentence $\langle s \rangle$ in the text expressing such relation". We adopt this hypothesis for both simple and complex relations (cf. infra), by considering the sentence $\langle s \rangle$ itself and the $[\langle s-k \rangle, \ldots, \langle s+k \rangle]$ adjacent ones, where k is a system parameter.

In order to identify sentences where one or more relations are expressed, we developed a system called *Text Frame Detector* (TFD).

Given a KB where domain terms are associated to a given set of frames, TFD populates them, by making explicit the semantic relation between terms and named entities (NEs). In particular, TFD exploits linguistic analysis and IE algorithms: texts are processed up to part of speech tagging, then NEs (Passaro et al., 2017) and multiword terms are identified (Passaro and Lenci, 2016). *Co-occurrency Analysis* (Asim et al., 2018) is then performed to identify the participants of each relation by considering terms and NEs co-occurring in the same sentence or in adjacent ones. The relations are filtered and ranked by applying a scoring process (cfr. Section 3.2) to the snippets containing them. The number of slots for each frame is not fixed, therefore we decided to store frames data in the graph-based database (GBD) Neo4j[1]. Compared to relational databases, GBDs do not require a pre-defined set of relations, allowing for a more flexible object-oriented data storage. Moreover, GBDs can be updated in real-time and show a better performance in terms of query execution time.

In order to increase the system recall of relevant information, we also used the semantic neighbors of the terms defining the frames. For example, if a text contains the word "versamento" ('deposit') but the KB only contains the word "pagamento" ('payment'), the term "versamento" may be extracted because it is a semantic neighbor of the latter (see Table 1).

Neighbor	Cosine Similarity
rimborso ('refund')	0.89
versamento ('deposit')	0.86
versare ('to deposit')	0.78

Table 1: Semantic neighbors of "pagamento" ('payment') and their cosine similarity score.

We trained *fastText* word embeddings (Bojanowski et al., 2017) on a combination of La Repubblica corpus (Baroni et al., 2004) and PAWAC (Passaro and Lenci, 2016) for administrative domain specific knowledge.

Currently, KB terms are expanded with their 10 nearest semantic neighbors in terms of cosine similarity, which can be filtered through a parametric threshold.

2.1 Definitions and terminology

Frame: Terms and entities contained in the KB are organized in frames. Frames allow to structure the implicit knowledge contained in texts around concepts that define the relevant semantic categories in a domain. For instance, the frame EVENT corresponds to en-

[1] http://neo4j.com/

tities like concerts, shows, etc. Each frame is defined by its *frame triggers* and *attributes*.

Frame trigger: It corresponds to an instance of the semantic class described by the frame (e.g., in the administrative domain, the frame TAX is expressed by its instances: "TARI" ('Garbage tax'), "IMU" ('Municipal tax')). Frame triggers suggest the presence of frame attributes in the text.

Attribute: A frame is composed by a set of slots, which must be filled by specific instances or data (Minsky, 1974). Each slot value is a participant in a relation with the frame trigger. This relation is referred to as an "attribute", and describes an aspect of the concept represented by the frame. For instance, the EVENT frame, requires the following attributes: **when**, to be filled with time and date, **where**, which corresponds to a location and **cost**, such as the ticket price. Depending on the way they are expressed in texts, we distinguish between *simple attributes* and *complex attributes*.

Simple attribute: Their values correspond to simple and complex terms, NEs or Temporal Expressions (TEs) identified during the IE step. The EVENT frame attributes are considered simple because they usually appear right near the frame trigger (cfr. Figure 1).

Complex attribute: The values of these attributes do not correspond to a single entity, but are expressed by whole text segments. Concerning the TAX frame, the **deadline** attribute cannot be filled by simply extracting the due dates from the text, because the reported information would be incomplete if taken out of context (cfr. Figure 2). Therefore, it is necessary to return the entire text snippet, which includes the *attribute triggers* that allow to identify the complex attribute.

Attribute trigger: They represent the linguistic cues of an attribute instance. They are manually selected by domain experts and stored in the KB with a standard form t and a small number of orthographic and morphosyntactic variants v. Attribute triggers can be: (i) single and multiword terms, like "bollettino postale" ('postal order'), "saldo" ('balance'), NEs, such as "Firenze" ('Florence') or TEs, like "18 giugno" ('18th June'); (ii) complex patterns, such as "non inferiore a" ('not lower than').

3 Implementation

In order to fill the frame slots, textual data are analyzed by TFD in various steps. After linguistic annotation, NER, and term extraction, TFD looks for frame triggers and for its attribute triggers, in the same sentence or in the sentences around it. More specifically, given a snippet , a frame instance F is expressed by a frame trigger F_t, and a set of attributes A, containing both simple (A_s) and complex (A_c) attributes, so that $F = \{F_t, A\}$ where $a_i \in A_s \cup A_c$.

3.1 Frame and attribute retrieval

Since both simple and complex attributes of a frame are expressed by means of the set T of their attribute triggers, we can say that F is instantiated in a text by the joint occurrence of a frame trigger F_t and a set of attribute triggers T related to one or more of its attributes, namely $F = \{F_t, T\}$ where $T = \{t1, ..., tn\}$.

In order to retrieve a frame F in a portion of text, first of all we look for its frame triggers. Once a F_t has been detected, we search for its potential attributes. Given such F, its potential instances in the text consist of the co-occurrence of F_t and a subset of T. To guarantee a certain degree of flexibility, we decided to provide each of the elements in T with a binary feature that can be set to 1 if the attribute trigger t_i is mandatory to extract the F, and to 0 if the attribute trigger is optional. A further implementation could consider to convert these features in continuous weights. In this way the TFD would be able to consider some triggers as more relevant than others to populate the frame.

Moreover, the attribute triggers of F belonging to T are selected within terms and entities used to express its attribute instances. Such triggers are then exploited by the attribute retrieval system of the TFD. Concerning the retrieval of simple attributes, see the extraction of the EVENT frame from the sentence in Figure 1.

The trigger for the EVENT frame ("spettacolo di Roger Waters") in Figure 1 is a clue for the presence of its attributes which populate the frame instance showed in Table 2.

Moreover, the TFD stores the raw text in Figure 1 as the relevant snippet for both the attributes

> Lo [spettacolo di Roger Waters]$_{nome\_evento}$ si terrà il [26 giugno]$_{data}$ allo [stadio di Firenze]$_{luogo}$.

Figure 1: Example of a snippet (‘Roger Waters’ show will take place on 26th June at the Florence Stadium’) containing simple attributes.

EVENT	spettacolo di Roger Waters
when	26 giugno
where	Stadio di Firenze
cost	-

Table 2: An instance of the EVENT frame.

when and **where**.

> Il [versamento]$_{pagamento}$ dell’[IMU]$_{tassa}$ deve essere effettuato con [bonifico bancario]$_{mod\_pagamento}$ o [bollettino postale]$_{mod\_pagamento}$ in due [rate]$_{somma}$: l’[acconto]$_{somma}$ entro il [18 giugno]$_{data}$ e il [saldo]$_{somma}$ entro il [17 dicembre]$_{data}$.

Figure 2: Example of a snippet (‘The Municipality tax disbursement must be made through wire transfer or postal order in two installments: down payment by June 18th and balance by December 17th) containing complex attributes.

Examples of complex attributes can be found in the TAX frame, namely **deadline**, indicating the due date of the tax payment, and **methods of payment**, indicating how it is possible to pay it. For example, the triggers detected for the attribute **deadline** in Figure 2 are “somma” (‘sum’), “pagamento” (‘payment’) and two TEs, namely “18 giugno” (‘June 18th’) and “17 dicembre” (‘December 17th’). The snippet contains also the attribute **methods of payment**, which is expressed by the triggers “pagamento” (‘payment’) and “mod_pagamento” (‘methods_payment’), expressed by “bonifico bancario” (‘wire transfer’) and “bollettino postale” (’postal order’). Table 3 shows the TAX frame instantiated with the extracted attributes. Also in this case, the full snippet (the raw text in Figure 2) is stored for both the attributes **deadline** and **methods of payment.**

3.2 Snippet selection and ranking

TAX	IMU
deadline	18 giugno, 17 dicembre
methods of payment	bonifico bancario, bollettino postale

Table 3: An instance of the TAX frame.

The binary features associated to each attribute trigger in a frame instance lead also the snippet selection and ranking system. Given a potential instance of a frame, its attribute triggers are associated with a binary feature indicating their compulsory presence in order associate the attribute with a certain snippet. On the basis of how many features are set to 1, the TFD will be more or less strict in the selection phase. For example, given the following sentences, where the frame triggers appear in bold and attribute triggers are underlined (the standard form for “pagata” is “pagamento” and “17 giugno” is marked as “data”), Table 4 shows which snippets are extracted according to the binary values associated to each attribute trigger.

A “L’**IMU** va pagata entro il 17 giugno” (‘The Municipality tax must be paid before June 17th’)

B “La scadenza dell’**IMU** è fissata al 17 giugno” (‘The deadline for the Municipality tax payment is on June 17th’)

Line	pagamento	scadenza	data	snippet
ID	(‘payment’)	(‘deadline’)	(‘date’)	extracted
1	0	0	0	A,B
2	0	0	1	A,B
3	0	1	0	B
4	0	1	1	B
5	1	0	0	A
6	1	0	1	A
7	1	1	0	-
8	1	1	1	-

Table 4: Mandatoriness of attribute triggers.

Each line of the table represents a potential combination of attribute triggers, with the respective mandatoriness. According to these features, the absence of mandatory attribute triggers (line 1) allows the retrieval of both the snippets A and B. Otherwise, if the system is expected to find all the attribute triggers (line 8), none of the two snippets is extracted because “pagamento” and “scadenza” never appear in the same sentence. This system is useful in order to balance the extraction flexibility based on the domain. For example, in administrative documents, where the language is bounded to stereotyped phrases (Brunato, 2015) a more strict approach is preferable, whereas in general domain ones it might be better to work with a higher number of optional triggers.

Moreover, a second objective of the TFD is to rank the extracted snippets according to their relevance with respect to a given attribute. Such relevance is calculated through a co-occurrence analysis, which employs measures based on semantic and distance features. One of these measures is the *Sentence score*, defined as:

$$SS = |t| \times |v| \tag{1}$$

where t is the number of attribute triggers (standard forms) and v is the total of their variants.

This formula takes into account the ratio between the number of attribute triggers and their variants. In particular, the TFD favours the snippets containing the highest number of distinct attribute triggers, namely their standard forms. In the case of simple attributes, t represents the number of entity types and v the number of NEs.

Furthermore, although different frame triggers may be found all over a given document, they may refer to the same domain entity, hence to the same frame instance. For example, we observed that Italian municipality web pages dedicate entire articles to a single tax, which can be mentioned in different ways, such as their full names and their acronyms (e.g., the Italian Tax “Imposta Municipale Propria” (‘Municipality tax’) can be mentioned also with the acronym, “IMU”). In order to avoid that attributes belonging to the same frame are associated to different ones and affect the scoring process, our system can be set to apply a “fuzzy normalization” strategy that is able to associate all the triggers of a document to a frame referring to the same entity. For example, the snippets extracted from a municipality web page and associated to the **deadline** attribute of the TAX frame can be ranked together, regardless the frame triggers they contain, such as “Imposta Municipale Propria” (‘Muncipality tax’) or its acronym, “IMU”.

At a document level, the snippet selected is simply the one with the highest *Sentence Score*, but we provide an additional level of analysis, which is applied when the snippet has to be chosen within a group of documents, instead of a single one. In that case, TFD selects the snippet with the highest *Document score* (DS), which encodes how likely the document contains a relevant information about a certain attribute. The Document score is calculated as follows:

$$DS = \frac{\sum_{i=1}^{n} TS}{l} \tag{2}$$

where l is the sentence length in terms of tokens, and TS is the *Trigger score* of a given variant v. TS is defined as:

$$TS = \frac{1}{d} \times cos \tag{3}$$

where d is the distance between the attribute trigger (or NEs) and the frame trigger, and cos is the cosine similarity between the trigger variant contained in the KB and the neighbor found in the text (the cosine is equal to 1 for the KB terms).

3.3 Storage

Extracted frame instances are stored in a Neo4j GDB. The Knowledge Graph (KG) contains several root nodes, one for each of the frames detected in the document or in the collection of documents (Figure 3).

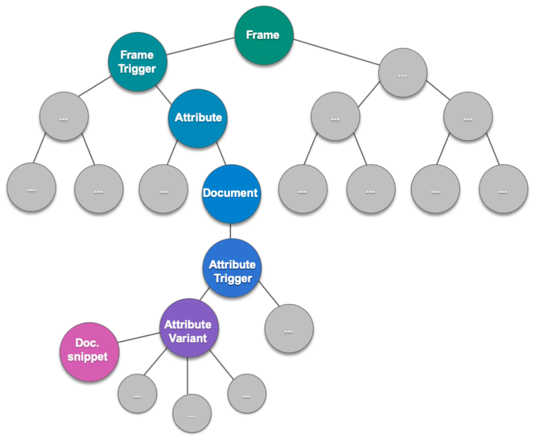

Figure 3: Information levels in the Knowledge Graph.

For instance, there are two root-nodes corresponding to the EVENT and TAX frames. If we consider the frame TAX (the node “Frame” in Figure 3), the nodes “Frame Trigger” can be populated with instances like “Imposta Municipale Propria” (‘Muncipality tax’) or its acronym, “IMU”. Each frame trigger node is linked to the corresponding frame attributes (“Attribute” node in Figure 3) which can be populated with information like “scadenza” (‘deadline’) and “modalità di pagamento” (‘methods of payment’). Document-nodes (“Document” node in Figure 3), labelled by document names, are placed between attribute-nodes and attribute-trigger-nodes in order to facilitate the retrieval phase. Each document node is associated with the snippet having the highest *Sentence score* for the connected attribute-node (e.g., 'deadline'), along with its *Document score*. In the retrieval phase, unless the information is extracted from a single document, the snippet with the higher *Document score* is selected and returned (see Section 3.2). The other levels of the graph contain information extracted from each document. Every attribute-trigger-node ("Attribute Trigger" node in Figure 3) is labelled by the standard form of the attribute trigger extracted from the connected document-node (e.g., 'sum'). Then, each attribute-trigger-node is connected to one or more nodes representing the trigger variants ("Attribute Variant" node in Figure 3). Continuing with this example, attribute variants can consist in 'installments', 'balance' and 'down payment'. Finally, the last node of the graph consists of the snippet-node ("Doc. snippet" node in Figure 3), storing the snippet containing the information extracted. For example, the node can be populated with a snippet like the one reported in Figure 2: "Il versamento dell'IMU deve essere effettuato con bonifico bancario o bollettino postale in due rate: l'acconto entro il 18 giugno e il saldo entro il 17 dicembre" ('The Municipality tax disbursement must be made through wire transfer or postal order in two installments: down payment by June 18th and balance by December 17th').

4 Evaluation and Results

The extraction of attributes related to TAX and EVENT frames were evaluated on Italian language texts by an administrative domain expert. We decided to evaluate these frames because the first one is very specific of the administrative domain, whereas the second one can be seen as a general purpose one. The gold standard includes both administrative documents as well as social media texts and news published on the municipalities websites. Both frames were evaluated on 50 texts, including information about taxes (municipality online guidelines), events (administrative acts, press releases, Facebook statuses and tweets) and other topics (municipality web pages). For municipality guidelines web pages, the "fuzzy normalization" strategy has been applied (see Section 3.2). The results of the TFD are shown in Table 5.

Frame	**Precision**	**Recall**	**F1**
TAX	0.771	0.519	0.621
EVENT	0.808	0.955	0.875
Total	**0.799**	**0.793**	**0.796**

Table 5: TFD evaluation results.

Since simple attribute values consist mostly of NEs, these results are strictly dependent on the generalization capability of the models used to extract those entities. In other cases, a wrong snippet is selected as relevant for an attribute, although triggers and NEs are correctly annotated and extracted. Moreover, additional errors depend on the absence of attribute triggers variants in the Knowledge Graph.

More specifically, errors are mainly related to a wrong NE annotation (35%). In the 22.8% of cases, a wrong sentence is selected as relevant for a certain attribute, although triggers and NEs are correctly annotated and extracted. False negative errors are caused by relevant information spread in several sentences (8.8%), whereas each extracted snippet consists of a single sentence, by unknown triggers describing an attribute (7.5%), by partial information contained in the extracted sentence (5%), by wrong lemmatization (1.75%) or by the overlapping of named entities and events (1.75%) (e.g., 'Roger Waters' show' is not annotated as an event, however 'Roger Waters' is extracted as a named entity). In other cases (3.5%), attribute triggers are too distant from their frame trigger to be extracted. Although this span is customizable, an excessive distance between frame and attribute triggers could produce noise in the retrieval phase. Finally, the application of the "fuzzy normalization" strategy (see Section 3.2) led to errors in the ranking phase (14.3%). One of the municipality web pages in which the strategy has been applied contained information on more than one tax, but only one frame instance has been returned. This kind of errors can be limited by automatically checking the frame triggers cited on the text, and deciding whether applying or not the normalization according to external lexical resources, such as gazetteers or dictionaries.

5 Conclusions

In this paper we presented a domain independent system for slot filling that exploits a graph to populate a frame-based ontology. The Text Frame Detector extracts a relevant snippet for each frame attribute from textual information with good results in terms of *F1 score* (0.796). Nonetheless, the evaluation showed that there is room for improvement in some of the TFD modules. For example, the annotation of the semantic neighborhood of single and multiword terms, which are particularly relevant in technical domains, should led to further improve recall performances for complex attributes.

Moreover, although we did not adopted Fillmore's semantic frames in the present work, we would like to explore the possibility of integrating our domain frames with FrameNet ones, which might contribute to enhance the system flexibility.

Finally, in the near future, we plan to fine-tune parameters and to implement additional features such as to associate multiple snippets to the same attribute. Furhermore, we intend to convert the binary features used in the snippet selection system into continuous weights. These weights, along with the collected data about frame population, would be also employed to train a supervised model for slot filling, in order to test TFD across new domains.

Acknowledgments

This research has been funded by the Project "SEM il Chattadino" (SEM), funded by Regione Toscana (POR CreO Fesr 2014-2020). The project brings together the CoLing Lab and the companies ETI3 s.r.l. (coordinator), BNova s.r.l. and Rigel Engineering s.r.l.

References

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings ACM 2000, the fifth conference of the Association for Computing Machinery on Digital libraries*, pages 85–94, New York, NY, USA.

Muhammad Nabeel Asim, Muhammad Wasim, Muhammad Usman Ghani Khan, Waqar Mahmood, and Hafiza Mahnoor Abbasi. 2018. A survey of ontology learning techniques and applications. *Database: the journal of biological databases and curation 2018*.

Marco Baroni, Silvia Bernardini, Federica Comastri, Lorenzo Piccioni, Alessandra Volpi, Guy Aston, and Marco Mazzoleni. 2004. Introducing the la repubblica corpus: A large, annotated, TEI(XML)-compliant corpus of newspaper Italian. In *Proceedings LREC'04, the fourth International Conference on Language Resources and Evaluation*, Lisbon, Portugal. European Language Resources Association (ELRA).

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5(Dec):135–146.

Dominique Brunato. 2015. *A Study on Linguistic Complexity from a Computational Linguistics Perspective. A Corpus-based Investigation of Italian Bureaucratic Texts*. Ph.D. thesis, Università di Siena.

Jinxiu Chen, Donghong Ji, Chew Lim Tan, and Zhengyu Niu. 2006. Relation extraction using label propagation based semi-supervised learning. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 129–136, Sydney, Australia. Association for Computational Linguistics.

Francesco Concoglioniti, Marco Rospocher, and Alessio Palmero Aprosio. 2016. Frame-based ontology population with pikes. *IEEE Transactions on Knowledge and Data Engineering*, 8(12):3261–3275.

Anthony Fader, Oren Etzioni, and Stephen Soderland. 2011. Identifying relations for open information extraction. In *Proceedings of EMNLP 2011. the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Edinburgh, Scotland, UK.

Charles J. Fillmore. 1976. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences: Conference on the origin and development of language and speech*, 280(1).

Aldo Gangemi, Mehwish Alam, Luigi Asprino, Valentina Presutti, and Diego Reforgiato Recupero. 2016. ramester: a wide coverage linguistic linked data hub. In *Proceedings European Knowledge Acquisition Workshop*, Cham. Springer.

Aldo Gangemi, Valentina Presutti, Diego Reforgiato Recupero, Andrea Giovanni Nuzzolese, Francesco Draicchio, and Misael Mongiovì. 2017. Semantic web machine reading with fred. *Semantic Web*, 8(6):873–893.

Ludovic Jean-Louis, Romaric Besançon, and Olivier Ferret. 2011. Text segmentation and graph-based method for template filling in information extraction. In *Proceedings of IJCNLP 2011, the fifth International Joint Conference on Natural Language Processing*, pages 723–731, Chiang Mai, Thailand.

Dan Jurafsky and James H. Martin. 2018. Speech and language processing. Third edition draft on webpage: `https://web.stanford.edu/~jurafsky/slp3/`. Accessed: 3 July 2019.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS 2013, 26th Conference* *on Advances in Neural Information Processing Systems*, pages 171–178, Lake Tahoe, Nevada, USA.

Marvin Minsky. 1974. *A framework for representing knowledge*. Massachusetts Institute of Technology, Cambridge, MA.

Mike Mintz, Steven Bills, Rion Snow, and Daniel Jurafsky. 2009. 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*, pages 1003–1011, Suntec, Singapore. Association for Computational Linguistics.

Raymond J. Mooney and Razvan C. Bunescu. 2005. Subsequence kernels for relation extraction. In *Proceedings of NIPS 2005, 18th Conference on Advances in Neural Information Processing Systems*, pages 171–178, Vancouver, British Columbia, Canada.

Thien Huu Nguyen and Ralph Grishman. 2015. Relation extraction: Perspective from convolutional neural networks. In *Proceedings of VS@NAACL-HLT 2015, the 1st Workshop on Vector Space Modeling for Natural Language Processing*, pages 39–48, Denver, Colorado.

Lucia C. Passaro and A. Lenci. 2016. Extracting terms with Extra. *Computerised and Corpus-based Approaches to Phraseology: Monolingual and Multilingual Perspectives*, pages 188–196.

Lucia C. Passaro, Alessandro Lenci, and Anna Gabbolini. 2017. Informed pa: A ner for the italian public administration domain. In *Proceedings of Clic-It 2017. The fouth Italian Conference on Computational Linguistics*, pages 246–252, Rome, Italy.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *Proceedings of ECML PKDD 2010, the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, pages 148–163, Barcelona, Catalonia, Spain. Springer.

Jacobo Rouces, Gerard De Melo, and Katja Hose. 2015. Framebase: Enabling integration of heterogeneous knowledge. In *Proceedings European Semantic Web Conference*, Cham. Springer.

Alisa Smirnova and Philippe Cudré-Mauroux. 2018. Relation extraction using distant supervision: A survey. *ACM Computing Survey*, 51(5):1–35.

Mihai Surdeanu. 2013. Overview of the tac2013 knowledge base population evaluation: English slot filling and temporal slot filling. In *Proceedings of TAC 2013, the Sixth Text Analysis Conference*, Gaithersburg, Maryland USA.

Daniel S. Weld, Raphael Hoffmann, and Fei Wu. 2008. Using wikipedia to bootstrap open information extraction. *SIGMOD record*, 37(4):62–68.

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

Meishan Zhang, Yue Zhang, and Guohong Fu. 2017. End-to-end neural relation extraction with global optimization. In *Proceedings EMNLP 2017, conference on Empirical Methods in Natural Language Processing*, pages 1730–1740, Copenhagen, Denmark.

Ming Zhenjun, Yan Yan Guoxin Wang, Janet K. Allen Joseph Dal Santo, and Farrokh Mistree. 2017. An ontology for reusable and executable decision templates. *Journal of Computing and Information Science in Engineering*, 17(3):031008.