



HAL
open science

Deep Learning and Natural Language Processing in the Field of Construction.

Rémy Kessler, Nicolas Béchet

► **To cite this version:**

Rémy Kessler, Nicolas Béchet. Deep Learning and Natural Language Processing in the Field of Construction.. 2025. hal-04882394

HAL Id: hal-04882394

<https://hal.science/hal-04882394v1>

Preprint submitted on 13 Jan 2025

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Deep Learning and Natural Language Processing in the Field of Construction.

Rémy Kessler and Nicolas Béchet

Université Bretagne-Sud
CNRS 6074A
56017 Vannes, France
`nicolas.bechet@univ-ubs.fr`

LIA / Université d'Avignon
339 chemin des Meinajariès,
84911 Avignon
`remy.kessler@univ-avignon.fr`

Keywords: terminology extraction, Internet queries, linguistic patterns, hypernym detection, embedding models, and deep learning.

Abstract. This article presents a complete process to extract hypernym relationships in the field of construction using two main steps: terminology extraction and detection of hypernyms from these terms. We first describe the corpus analysis method to extract terminology from a collection of technical specifications in the field of construction. Using statistics and word n-grams analysis, we extract the domain's terminology and then perform pruning steps with linguistic patterns and internet queries to improve the quality of the final terminology. Second, we present a machine-learning approach based on various words embedding models and combinations to deal with the detection of hypernyms from the extracted terminology. Extracted terminology is evaluated using a manual evaluation carried out by 6 experts in the domain, and the hypernym identification method is evaluated with different datasets. The global approach provides relevant and promising results.

1 Introduction

The current era is increasingly influenced by the prominence of smart data and mobile applications. The work presented in this paper has been carried out in one industrial project (VOCAGEN) aiming at automating the production of structured data from human-machine dialogues. Specifically, the targeted application drives dialogues with people working in a construction area for populating a database reporting key data extracted from those dialogues. This application requires complex processing for both transcribing speeches and driving dialogues. The first process ensures good speech recognition in noisy environments. The second processing step is necessary to ensure the database contains accurate

and complete data; indeed, people tend to apply a broad (colloquial) vocabulary and the transcribed words must be used to point to the correct information. Additionally, if some data populate the database, additional data may be required for completeness, thus the dialogue should enable getting those additional data (e.g. if the word “room” is recognized and used to populate the database, the location of the room must also be obtained; this can be done by driving the dialogue).

The application provides people with a “hands-free” device, enabling complete, quick, and standardized reporting. The first usage of this application will be oriented to reporting failures and problems in construction site.

The two processing steps mentioned above require on the one side a “language model” (for transcribing the sentences) and on the other side a “knowledge model” for driving the dialogue and correctly understanding the meaning of the word. The knowledge model is mainly an ontology of the domain (in this case, the construction domain) providing standardized concepts and their relationships. As well known, building such knowledge models takes time and is costly; one of the earlier questions raised by our industrial partners has been about “how to build, as automatically as possible, such a knowledge model”. This question is closely related to the interest in quickly adapting the application to other domains (than the construction one) for reaching new markets. We developed a complete methodology and system for partially answering the question, focusing on how to extract relevant terminology from a collection of technical specifications. The rest of the paper is organized as follows. Section 2 presents the context of the project. Related works are reviewed in Section 3. Section 4.1 presents collected resources and some statistics about them. Section 4.2 describes the methodology developed for extracting relevant terms from collected resources. The details about the evaluation are presented in Section 4.3 and the obtained results are given in Section 4.3. Section 5 provides details about how processing is introduced for facing the problems of detection of hypernyms and describes different strategies tested. The details about the evaluation are presented in Section 5.2 and the obtained results are given in Section 5.2.

2 Industrial context

Figure 1 presents the context of this work in VOCAGEN project. Our industrial partner, Script & Go¹, developed an application for construction management dedicated to touch devices and wishes to set up an oral dialogue module to facilitate the construction sites data collect. The second industrial partner, Tykomz, develops a vocal recognition suite based on the Sphynx 4 toolkit (Meignier and Merlin, 2010, 10-19). This toolkit includes agglomerative hierarchical clustering methods using well-known measures such as BIC² and CLR³ and provides elementary tools, such as segment and cluster generators, decoders and model

¹ <https://scriptandgo.com/fr/>

² the Bayesian information criterion (BIC)

³ Cross Likelihood Ratio

trainers. Fitting those elementary tools together is an easy way of developing a specific diarization system.

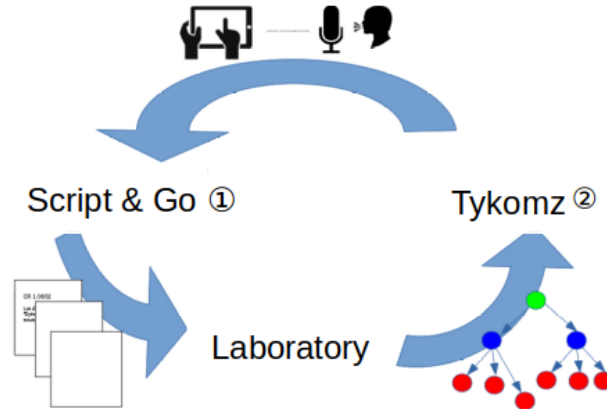


Fig. 1. figure describing the context of the project

For this application to work, it is necessary to build a model of knowledge, i.e. a model describing the expressions that must be recognized by the program. To improve the performance of the system, this knowledge model must be powered by a domain-specific vocabulary. For example, in the sentence “There is a stain of paint in the kitchen”, the system must understand that it is a stain of paint and that the kitchen is a room. Figure 2 presents the final objective of the VOCAGEN project and describes connections between each partner. In the first step, the user downloads the software from Script & Go (①) onto their smartphone or tablet which is connected to the partner’s server for requests, logging, and updates. Users can use the hands-free module and the software then connects to the voice recognition engine of the second partner to analyze the voice message and transmit a transcription. To perform this transcription, this voice recognition engine uses a language model and a base of test voice messages (②). In case of an incorrect voice transcription, the user can manually correct the transcription, and the unrecognized sentences are preserved and analyzed by another process to be able to generate a model allowing these errors to be corrected (③). After a step of manual evaluation of the quality of the new language model, a new APK⁴ is generated so that the user can download a new version (④).

To our knowledge, there is no ontology or taxonomy specific to the construction industry in French. A version is under development by (Pauwels and Terkaj, 2016, 100-133) but the ontology is in English and very generic. We therefore choose to extract useful knowledge from textual data, and then, in a second step, organize it.

⁴ Android Package (or APK, for Android Package Kit) is a file format for the Android operating system.

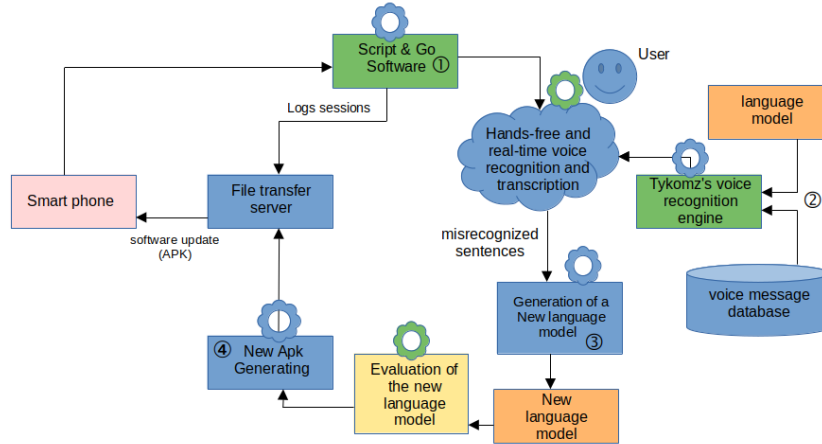


Fig. 2. figure describing the final process chain

3 Related works

The goal of ontology learning (OL) is to build knowledge models from text. OL uses NLP knowledge extraction tools to extract terminology and links between them (relationships). It is concerned with discovering knowledge from various data sources and representing them in an ontological structure. Thus ontology learning comprises a set of techniques to extract the core components of the ontology, i.e. concepts, taxonomic and ad hoc relations, and general axioms. For more details, (Asim et al., 2018, 1-24) provide an overview of existing techniques to accomplish the various subtasks of ontology learning.

3.1 Taxonomy Extraction

Our interest in this part of the work was focused on extracting taxonomic relations. A taxonomic relation occurs between two concepts where a concept is a superordinate of another concept. For example, the concept “Fish” is a superordinate of the concept “Shark” (is-a(Shark, Fish)). We base our approach on the fact that the hypernyms (i.e. relations between terms - a term may comprise several words - in a text representing the fact that a term can be used instead of another term in a sentence, conveying more general or more specific meaning) suggest taxonomic relationships. The reference in the field of rule-based systems was developed by (Cunningham, 2002, 223-254). General Architecture for Text Engineering (GATE) is a Java collection of tools initially developed at the University of Sheffield in 1995.

An alternative is offered by the existing semi-automatic ontology learning system `text2onto` (Cimiano and Völker, 2005, 227-238). More recently, (Kluegl et al., 2016, 1–40) developed UIMA, a system that can be positioned as an alternative to GATE. Amongst other things, UIMA makes possible to generate rules from a collection of annotated documents. `Exit`, introduced by (Roche and Kodratoff, 2004, 946-956) is an iterative approach that incrementally finds the terms.

(Biébow et al., 1999, 49-66) with TERMINAE is certainly the oldest statistic approach. Developed for French and based on lexical frequencies, it requires pre-processing with `TermoStat` (Drouin, 2003, 99-115). (Chowdhury et al., 2018, 1-14) presents a method for extracting terminology specific to a domain from a corpus of domain-specific text, where no external general domain reference corpus is required. They present an adaptation of the classic *tf-idf* as a ranking measure and use different filters to produce a specific terminology. More recently, the efficiency of ranking measures like mutual information developed for the statistical approach is discussed in (Bouma, 2009, 31-40) and (Bestgen, 2017, 10-19). (Meyers et al., 2018, 1-14) proposes `Termolator`, a terminology extraction system using a chunking procedure, and using internet queries for ranking candidate terms. The approach is interesting but the authors emphasize the fact that the runtime for each query is a limiting factor to produce a relevant ranking.

Closer to our work, (Gillam et al., 2005, 55-81) presents an approach combining linguistic patterns and Z-score to extract terminology in the field of nanotechnology. (Panchenko et al., 2016, 1320-1327) propose TAXI, which combines statistics and learning approaches with corpus comparison like `Termostat`. TAXI is a system for building a taxonomy using 2 corpora, a generic and a specific. It ranks the relevance of candidates by measure (frequency-based), and by learning with SVM. (Lefever et al., 2009, 496-504) and (Macken et al., 2013, 1-30) present `TexSIS`, a bilingual terminology extraction system with a chunk-based alignment method for the generation of candidate terms. After the corpus alignment step, they use an approach combining log-likelihood and Mutual Expectation measures (Dias and Kaalep, 2003, 81–91) to rank candidate terms in each language. Similarly, (Daille, 2002, 1-8) and (Lang et al., 2018, 427-472) present an approach to extract grammatical terminology from linguistic corpora. They compare a series of well-established statistical measures that have been used in similar automatic term extraction tasks and conclude that corpus-comparing methods perform better than metrics that are not based on corpus comparison. (Amjadian et al., 2016, 2-11) and (Wohlgenannt and Minic, 2016, 1-4) present methods with words embedding. With a small data set for the learning phase, they improve the term extraction results in quality of n-gram extracted. However, these papers involve labeled data sets for the learning phase, which is the main difference with our proposed approach. Our approach combine a lexico-syntactical and a statistical approach while using external resources.

3.2 Detection of hypernyms

Ontology learning refers to the automatic or semi-automatic building of ontology. It is concerned with discovering knowledge from various data sources and representing them in an ontological structure. Thus ontology learning comprises a set of techniques to extract the core components of one ontology i.e. concepts, taxonomic and ad-hoc relations, and general axioms. For more details, (Asim et al., 2018, 1-24) provide an overview of existing techniques to accomplish the various subtasks of ontology learning.

The main approaches found in the literature for extracting hypernyms are pattern-based and distributional, being the latter further divided into unsupervised and supervised. Pattern-based approaches are heuristic methods that predict hypernym relations between pairs of terms if those terms are related by one pattern matching with a given sentence. These patterns are either defined manually or extracted automatically. The earliest and most popular handcrafted patterns are introduced by Hearst (Hearst, 1992, 1-7), thus known as Hearst’s patterns. Pattern approaches can be used for detecting hyperonyms in a text: this is not the case in the work presented in this paper because we extract hypernyms by using a terminology previously extracted and validated.

Distributional-based approaches are based on the distributional hypothesis, suggesting that words sharing the same linguistic context tend to have a similar meaning (Harris, 1968, 1-6). Earlier works to predict hypernymy are unsupervised, usually based on symmetric measures such as cosine similarity (Salton and McGill, 1986, 1-15) and Lin similarity (Lin, 1998, 296-304). Various evolutions of earlier works have been proposed such as (Weeds et al., 2004, 1015-1021), (Shwartz et al., 2016, 425-435), (Roller et al., 2014, 1025-1036), enabling to take into account the inclusion of the context of each term in a pair.

(Lenci and Benotto, 2012, 75-79) propose to take into account both the inclusion and the non-inclusion of the context of each term in a pair. Unsupervised methods are simple to implement and apply and there is no need for training data: however, they show low performance and are heavily domain-dependent.

Supervised learning approaches rely on a training dataset to train a model. The model is then used to predict hypernym relations between terms in a pair. Most of these approaches are based on words embedding like Word2vec (Mikolov et al., 2013, 3111-3119) or Glove (Pennington et al., 2014, 2249-2259). (Levy et al., 2015, 970-976) present a study of supervised approaches on standard corpora in the field and compare state-of-the-art results on labeled datasets. The various supervised methods differ in the way they represent each candidate pair of words (x, y) : In (Weeds et al., 2014, 2249-2259) and (Roller et al., 2014, 1025-1036), they use the difference between the embedding vector of terms y and the embedding vector of terms x ($y - x$) as a feature vector to train an SVM classifier. They conclude asymmetric representation performs better and the difference representation yields the best result. (Luu et al., 2016, 403-413) proposed an approach to encode hypernym properties by learning terms embedding that not only indicate the information of the hyponym and the hypernyms but also the contexts between them. Consequently, they define triples of hypernym, hy-

ponyms, and the context words between them. They use these triples as training data to build an SVM model by using as feature vectors the concatenation of the embedding vector of hyponyms, the embedding vector of hypernym, and the difference between the two vectors ($x \oplus y \oplus y - x$).

According to (Mirkin et al., 2006, 579-586), the pattern-based and distributional approaches have certain complementary properties and they propose a combined approach by learning a supervised model using a set of features obtained by concatenating pattern-based features and distributional-based features. In a later work, (Shwartz et al., 2016, 425-435) also proposed an approach that combines pattern-based and distributional methods with a supervised model where they learn a neural network model using a concatenation of three feature vectors. A pattern-based feature vector represents the dependency path occurrences of a term pair (x, y) and the other two feature vectors are distributional-based vectors and they represent the embedding vectors of x and y .

More recently, pre-trained large-scale language models, such as BERT (Devlin et al., 2019, 4171-4186), RoBERTa (Liu et al., 2019, 1-13), and CamemBERT (Martin et al., 2020, 7203-7219) for French, have proven to be highly effective for many tasks in NLP. For example, (Vetter et al., 2022, 282-290) and (Markchom et al., 2022, 260-265) propose two similar approaches of fine-tuning the BERT-based models for taxonomic relation classification to tackle one task similar to our at SemEval 2022⁵. Recent works on large-scale pre-trained language models (LLM), such as GPT-3 (Brown et al., 2020, 1877-1901), Llama-2 (Touvron et al., 2023, 510-521) and Claire-Mistral (Hunter et al., 2023, 1-11), suggest that LLMs also perform well in various NLP tasks. Large Language Models (LLMs) feature powerful natural language understanding capabilities. With only a few examples, an LLM can be prompted to perform custom NLP tasks such as text categorization, named entity recognition, information extraction and more. Close to our task, (Ma et al., 2023, 5858-5867) propose a method for Chinese hypernym relation detection and use the concept of prompt learning to incorporate prior knowledge in the form of patterns into the model.

Although supervised methods overall show better performance than unsupervised ones, they are heavily dependent on the quality and size of the dataset. For our purpose, we decided to use supervised methods because they are currently easy to implement, language-independent, transferable to other domains, and compatible with our overall approach where only terminology (not complete texts) is used for detecting hypernyms. By varying approaches, therefore we analyzed which of the state-of-the-art methods performs better in our case.

⁵ SemEval is a series of international NLP research workshops focus on semantic analysis <https://semeval.github.io/>

4 Terminology Extraction

4.1 Resources and statistics

The first experiments were carried out using technical reports⁶ collected from three customers from our industrial partners who will be called NC⁷ collection thereafter. Each document contains all the non-compliance that was found on one work site and describes solutions to resolve it. However, the heterogeneity of the formats, as well as the artificial repetition of the information between two reports found on the same construction site, made the term extraction quite difficult. A detailed analysis of these reports reveals a rich vocabulary. However, the presence of numerous misspellings, typing shortcuts, a highly telegraphic style characterized by infinitive verbs, minimal punctuation, and a scarcity of determiners significantly hinders the exploitation of this vocabulary. As a consequence, we used a collection of technical specifications called CCTP⁸. CCTPs are available online on public sector websites⁹. Several thousand documents were collected by our industrial partner using an automatic web-collecting process. Figure 3 presents some key descriptive statistics of these collections.

Collection	NC	CCTP
Total number of documents	58 402	3665
<i>Without pre-processing</i>		
Total number of words	130 309	230 962 734
Total number of different words	93 000	20 6264
Average words/document	125.3	63 018.48

Fig. 3. statistics of the collection.

4.2 Methodology of Extraction of Terminology

System Overview Figure 4 presents an overview of the system designed and implemented: steps are explained in further sections. In Step 1 pre-processing of raw information extracted from CCTP collection takes place; this is required for normalizing the entire set of documents. In Step 2, n-grams are extracted (by using measures). 1,2,3 grams are extracted. In Step 3, n-grams are filtered by using linguistic patterns and Internet queries. Finally, in Step 4 a ranking is applied to the filtered n-grams.

⁶ Site meeting report

⁷ For Non-Compliance

⁸ The technical specifications book (CCTP in French) is a contractual document that gathers the technical clauses of a public contract in the field of construction.

⁹ For example, <https://www.marches-publics.gouv.fr/> or <http://marchespublics.normandie.fr/>.

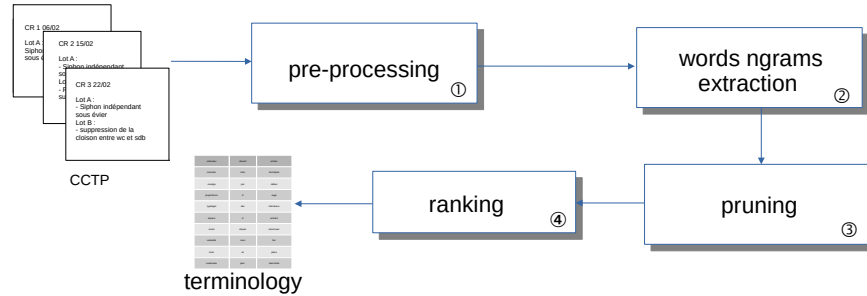


Fig. 4. System overview

Normalization, pre-processing and word n-grams extraction In Step 1, a text normalization is performed to improve the quality of the process. We remove special characters such as “/” or “()”. Different pretreatments are done to reduce noise in the model: we remove numbers (numeric and/or textual) and special symbols. “.” are tagged with a special character to not create artificial n-grams. Specific words (including named entities) like company names, dates, etc. are normalized and will be removed in the next module. We do not include a stop list to keep n-grams with prepositions, for the purpose described in the remainder. Then, we tokenize the entire collection before using TreeTagger Schmid (1994) to get the part-of-speech tags and lemmas of each word. After this step we transform all vocabulary from the CCTP collection into 1-grams, 2-grams and 3-grams. Special characters or normalized words resulting from the previous processing are discarded. N-grams with a very low frequency (2) are also discarded.

Linguistic patterns module We use grammatical labels generated in the previous step (section 4.2) and linguistic patterns to retrieve collocations such as NOUN-NOUN and NOUN-PREP-NOUN. These patterns are frequently found in the literature (Roche and Kodratoff, 2004, 946-956) to capture specific words in French like “carte de crédit” (*credit card*) and discard 3-gram like “créditer sa carte” (*credit his card*) with the pattern VERB-PREP-NOUN. Among frequent patterns found in literature, those patterns have been selected according to the statistics obtained from a knowledge model of another field (agriculture), given by one of our industrial partners. Figure 5 presents the main patterns we selected using this knowledge model. The sum of percentages may not reach 100% as infrequent patterns are excluded. We observed that the noun-based patterns are the most frequent patterns, whatever the size of the n-gram. The other selected patterns also contain nouns, but they are n-grams with verbs, adjectives or

	Number	Percentage
1-grams	1360	65.24%
NOUN	1037	76.25%
VERB	194	14.26%
ADJ	120	8.82%
2-grams	390	19.57%
NOUN-NOUN	346	88.72%
ADJ-NOUN	11	2.82%
PREP-NOUN	7	1.79%
VERB-NOUN	5	1.28%
3-grams	188	9.43%
NOUN-NOUN-NOUN	150	79.79%
PREP-NOUN-NOUN	15	7.98%
NOUN-PREP-NOUN	6	3.19%
VERB-NOUN-NOUN	6	3.19%

Fig. 5. Distribution of linguistic patterns according to the knowledge model.

prepositions. Therefore, we have configured our system to keep only the n-grams corresponding to these patterns.

Pruning step This step uses the Internet to prune n-grams for which no information is returned after querying Bing¹⁰ search engine. We count the number of links in the result pages that contain the n-gram exactly. We save the number of exact matches between the n-gram and the title and snippet of each result. We keep only the n-grams whose number of matches exceeds a defined threshold. We varied this threshold between 1 and 50 and results presented in Section 4.3 have been obtained with a threshold empirically set to 10.

Ranking step We tested several measures as provided in (Roche and Kordatoff, 2004, 946-956) and (Macken et al., 2013, 1-30) like mutual information in order to rank selected n-grams by quality but the results were disappointing. We finally use classical *Z score* (Altman, 1968, 589-609) with twenty years of the French newspaper Le Monde¹¹ as a generic collection. This metric considers word frequencies weighted over two different corpora, in order to assign high values to words having much higher or lower frequencies than expected in a generic collection. We defined it as follows :

$$p_1 = a_0/b_0 \quad (1)$$

$$p_2 = a_1/b_1 \quad (2)$$

$$p = (a_0 + a_1)/(b_0 + b_1) \quad (3)$$

¹⁰ <https://www.bing.com/>

¹¹ <http://www.islrn.org/resources/421-401-527-366-2/>

$$Z_{score} = \frac{p_1 - p_2}{\sqrt{(p * (1 - p) * (\frac{1}{b_0} + \frac{1}{b_1}))}} \quad (4)$$

Where a is the lexical unit considered (1-gram, 2-gram or 3-gram), a_0, a_1 the frequency of a in the CCTP collection, b_0 the total size in words of CCTP collection, b_1 the frequency of a in the collection Le Monde.

4.3 Experiments and results

Experimental protocol To assess their quality, we manually evaluated all 3-grams the system retained. Six construction specialists participated, each evaluating approximately one third of the data. 5144 3-grams were evaluated with this method and each n-gram was evaluated by 2 different specialists¹². For each n-gram, the specialist can choose between three possibilities:

1. the 3-gram is irrelevant ;
2. the 3-gram is relevant but does not belong to the domain¹³ ;
3. the 3-gram is relevant and belongs to the domain.

The evaluation was done in two steps and we use Kappa measure¹⁴ (Cohen, 1960, 1-10) and inter-annotator agreement at the end of the first step to show the difficulty of the task. At the end of the first step, we obtained a Kappa score of 0.62 and a global inter-annotator agreement of 0.74, which is quite good as explained in (McHugh, 2012, 1-7). The difficulty of the task was to distinguish the domain-specific vocabulary from the generic vocabulary used in the field of construction. Each disagreement was re-evaluated in the second step by a pair of experts. Figure 6 shows the final results of the evaluation.

Results In this section, we present the results obtained during the manual evaluation of the 3-grams retained by the system. We only compute the accuracy and the error rate, because we are not able to compute the recall for this collection¹⁵. We have merged the assessments of each expert using two different evaluation rules:

- a strict evaluation where an n-gram is considered correct if both experts have rated it relevant and in the domain.
- a flexible evaluation where an n-gram is considered correct if both experts consider it relevant and at least one of the experts considers it the domain.

¹² With a inter-annotator agreement of 0.83

¹³ For example, credit card (carte de crédit in french) is a relevant n-gram but does not belong to the domain of construction

¹⁴ We use general formula as follows: $\kappa = \frac{A_0 - A_e}{1 - A_e}$ where A_0 = observed agreement and A_e = expected (chance) agreement.

¹⁵ Indeed, we do not know every relevant terms existing in the corpus, so we cannot estimate the recall for the collection of terms we automatically extract.

	strict evaluation	flexible evaluation
accuracy	0.77	0.91
error rate	0.23	0.09

Fig. 6. Results of manual evaluation on the 3-grams.

The strict evaluation shows good quality results (0.77). Analysis of the results shows that the main error is related to “incomplete n-grams”. For example, the 3-gram “personne à mobilité” (person with mobility) is not relevant while the 4-gram “personne à mobilité réduite” (person with reduced mobility) can belong to the field of construction. Some errors can also be traced back to the CCTP documents. For example, “engin de guerre” (war machine) is a term that does not belong to the field but a law relating to the presence of a war machine on the building sites is reported in every CCTP. The flexible evaluation shows very good results (0.91) and the difficulty of assessing the class of some terms such as “absence de remise” which has 2 distinct meanings in French (no outhouse and no discount). The first meaning is relevant in the field of construction but not the second. This part of the work was previously published in (Kessler et al., 2019, 22-26). After the terminology extraction step, we worked on a model to detect hypernym relationships, because this task was far too time-consuming to do manually with all the extracted terms.

5 Detection of Hypernym Relationships

This section focuses on relation extraction from the terminology previously obtained. Two methodologies are studied, the first use classical words embedding vectors as features, and the second one relies on end-to-end approaches. To validate the methodology, two other datasets are experimented with.

5.1 Methodology

Words embedding-based module

Overview Figure 7 presents the architecture of the Words embedding-based module. In the first step, we use a training dataset (①) composed of pairs of terms that are known to be in a hypernym relationship (or hyponym relationship depending on the reading order). For instance, the pair “kitchen, room” represents a hyponym relationship because a kitchen is a room. We combine this training dataset with a words embedding model (②) providing a vector for each term of each pair in the training dataset. A vector algebraic operation (③) is then applied between each word of the pair to create a hypernym relationship learning model (④). Such kind of model is fit using classical machine learning algorithms like Random Forests or Multilayer Perceptron. Finally, the obtained model is used to detect hypernym relationships in the extracted terminology.

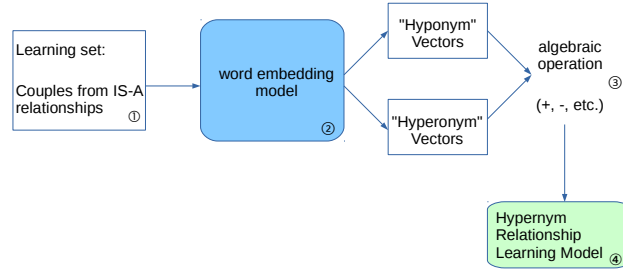


Fig. 7. Architecture of the hypernym detection module

Embedding models component Words embedding is a representation technique to represent any word in the low-dimensional space: words having similar representations are likely to be semantically similar. Each word found in an input dataset is projected to a vector model to obtain a semantic representation of the whole input dataset. We experimented with three words embedding models as follows. The first model is Word2vec (Mikolov et al., 2013, 3111-3119), an unsupervised and predictive neural words embedding technique to learn the word representation in low-dimensional space. We specifically use the skip-gram model, in which a word is used to predict the context using a neural network close to an autoencoder. Second is Glove, for Global vector for Word Representation (Pennington et al., 2014, 1532–1543). A co-occurrence word matrix is created from a text dataset for the training and is reduced in low-dimensional space which explains the variance of high-dimensional data and provides a word vector for each word. The last model is fastText (Bojanowski et al., 2017, 135-146), which is close to Word2vec. Unlike Word2vec which considers each word as a single unit and ignores the morphological structure of the word, fastText overcomes this limitation by considering each word as an n-gram of characters.

Note that we do not mention in our experimental section the results obtained with Word2vec because they were systematically lower than those obtained with fastText.

Vector compositions and machine learning algorithms We tested four compositions for representing (x, y) as a feature vector: **concat** ($x \oplus y$) Baroni and Lenci (2010), **diff** ($y - x$) Weeds et al. (2014), **sum** $v_b + v_a$ and **product** $v_b * v_a$. We evaluated the performance of the following classification algorithms: SVM (Support-Vector Machine), RF (Random Forests), and MLP (Perceptron Multilayer), and two fusions of vote-based algorithms: the fusion so-called Hard (the majority prevails) and the so-called Soft (we sum the probabilities of pre-

diction). However, we have got bad results with fusion and have not included results in the paper.

End-to-end based module

Overall Unlike the module presented previously based on classic words embedding, an end-to-end approach can be defined generically as a system that processes the entirety of a task in a manner unified, without manual or decoupled intermediate steps. In effect, the system takes raw data as input and produces the desired result, without requiring human interventions or third-party systems with the possible exception of minimal pre/post-processing. Usually, we use a pre-trained model to implement an end-to-end approach.

For our task, such an approach amounts to using two words as input to the model (the hyponym and the hypernym) and as the expected output the class label (i.e. in relation of hyperonymy or not). Thus, the model can be fine-tuned with the training dataset. We do not need to use precomputed embedding like Word2vec because the model will learn contextualized embedding weights during the training process.

Used algorithms We evaluated the performance of three algorithms which can be used in an end-to-end process. A mask-based algorithm RoBERTa (and this French version CamemBERT), and two causal models (which are also Large Language Models) Llama-2 and Claire-Mistral.

RoBERTa (Robustly Optimized BERT Pretraining Approach) is a pre-trained Transformers-type language model, developed by Facebook in 2019. It is an optimized and improved version of the BERT model, with pre-training on many more data.

LLaMA-2 (Longform Language Model with Attention) is a Large Transformers-type Language Model, recently developed by Facebook in 2023. We use here the 7B model.

CLAIRE/Mistral is a Large Language Model developed by French researchers in 2023 adapted from Mistral. Built on the Transformers architecture, it was pre-trained specifically on diarized French conversations.

Experimental protocol and results are presented in the next section.

5.2 Experiments and results

Datasets We rely on three datasets of semantic relations, which are all used in various state-of-the-art approaches, for hypernym’s evaluation. The first dataset is BLESS (Baroni et al., 2012, 23-32), which contains 1,337 hyponym-hypernym pairs. It is designed to evaluate distributional semantic models. It contains 200 distinct concepts. Concepts are named with single-word nouns in the singular form. Each concept has a set of related words. A concept and related word are labeled by one of the following five relations: co-hyponym, hypernym, meronym, attribute, and events. For instance, the dataset contains “cat-hyper-animal” where

“cat” is the concept, “animal” is the related word, and “hyper” is the label. The second data dataset is EVALution, introduced by (Santus et al., 2015, 64-69). It is also designed for the purpose of training and evaluating distributional semantic models. It consists of 7,500 couples representing the following semantic relationships: hypernymy, synonymy, antonymy, and meronymy. Couples were extracted from ConceptNet 5.0 and WordNet 4.0 (Liu and Singh, 2004, 211-226). The latest dataset, VOCAGEN, written in French, contains 4,143 couples, specific to the construction domains, extracted from the Tykomz partner knowledge model.

For each dataset, we define, if not available, *positive couples* (the ones containing hyponymy relationships), and *negative couples* (the ones not containing hyponymy relationships). BLESS dataset already contains negative couples. For the two other datasets, we use the following process to define negative couples. We first randomly select a list of hypernyms noted H from the positive couples. Then, we add to each H element a fixed n number of words that we did not get with the hypernym before, in order to be sure to produce a negative couple. In the end, we fix the value of n in order to have a balanced dataset (the same number of positive and negative couples).

Words embedding used models We have used generic models for English and French. We originally built a specific model based on technical documents collected by our industrial partner using an automatic web-collecting process and described in section 4.1, but the results were very disappointing and were not included in the paper.

Pre-trained embedding models for English datasets BLESS and EVALution are the following.

- Two GloVe models are used: **glv_core**¹⁶ and **glv_vect**¹⁷. These models are included in the SpaCy python module. `glv_core` is the embedding model from SpaCy called “en_core_web_lg”. This model was trained with GloVe from the Common Crawl dataset (<https://commoncrawl.org/>). `glv_vect` is also trained with GloVe from blogs, news, and comments.
- The **fastText** vectors (Mikolov et al., 2018, 1-4) contain 2 million word vectors trained on Common Crawl and are available on the official web page of the authors¹⁸.

For the VOCAGEN dataset, we use the French version of these pre-trained embeddings.

Machine learning used models We use the scikit-learn¹⁹ implementation for classical machine learning algorithms SVM (Support-Vector Machine) RF (Random Forests), and MLP (Multilayer Perceptron). We also experiment TPOT²⁰

¹⁶ https://spacy.io/models/en#en_core_web_lg

¹⁷ https://spacy.io/models/en-starters#en_vectors_web_lg

¹⁸ <https://fasttext.cc/docs/en/crawl-vectors.html>

¹⁹ <https://scikit-learn.org/>

²⁰ <https://epistasislab.github.io/tpot/>

which is an auto-classification models tool that combines different statistical analyses and machine learning algorithms to get the best possible model. The used parameters for these different algorithms are the following:

- MLP: We use two layers of size 100 with a ReLU activation, an Adam solver, a batch size of 32 with a maximum iteration number of 100, a random state of 42, and an early stopping with a tolerance of 0.0001.
- RF: We use 1,000 estimators, a max depth of 1,000, and a random state of 42.
- SVM: meta parameters of SVM are fixed using a cross-validated grid search on the training dataset. Tested parameters values are C: .1, 1, 10, 100, 1,000; gamma: (scale, auto); and kernel: rbf, sigmoid.
- TPOT: We used pre-implemented scikit-learn algorithms with TPOT with the following parameters: number of generations: 10, a population size of 100 and a random state of 42.

In the case of other models, we use the Hugging Face²¹ implementation and the `AutoModelForSequenceClassification` class to add an appropriate output head to the model. For CamemBERT/RobERTa, we use the base and large versions of the models. RobERTa is used for English datasets and CamemBERT for the French dataset. The following parameters are used: a learning rate of 5e-5, a batch size of 64, and a maximum of 100 epochs (using an early stopping patience of 5). Other parameters are the default ones. With the large version of these models, we change the learning rate to 1e-6. We experiment with the `Llama-2-7b-hf` and the `Claire-Mistral-7B-0.1` models. The following parameters are used: a 4-bit quantization is made by using the `bitsandbytes` module, adding an adapter for fine-tuning using Lora with `r=16`, `lora_alpha=32` and all target modules. We use a batch size of 16 and a maximum of 100 epochs (using an early stopping patience of 5). The learning rate is fixed to 2e-5.

Experimental protocol The experimental protocol is as follows. The objective is to check the quality of the prediction model for each selected dataset. Only the positive class is evaluated, that is to say that we only search if the couples judged to be hypernym relations are so. Each of the experiments is evaluated using the well-known measures of Precision, Recall, and F-measure, averaged over all classes (with $\beta = 1$ in order not to privilege precision or recall (Goutte and Gaussier, 2005, 345-359)). The results were obtained using 5-fold cross-validation on each dataset²². The classifier performance is shown in the table below. The table includes the average F1-score across the 5 folds (mean) and the standard deviation (\pm SD) for each measure. Results are finally compared to a baseline model.

Results

²¹ <https://huggingface.co/>

²² We use this distribution of data : BLESS train 1000, dev 334, test 334, EVALution : train 2256, dev 752, test 752

State-of-the-art datasets results Figures 8 and 9 present a summary of the results obtained with each classification algorithm and each model on both BLESS and EVALution datasets.

Algo	Emb/Size	μ F1	μ R	μ P
Llama-2	7B	97.81 \pm 1.14	98.80 \pm 0.93	96.86 \pm 1.84
RF	FastText	97.55 \pm 0.90	95.69 \pm 1.98	99.51 \pm 0.46
SVM	Glove	97.54 \pm 0.42	97.48 \pm 0.70	97.62 \pm 1.10
Mistral	7B	97.31 \pm 0.64	97.60 \pm 0.66	97.04 \pm 1.21
TPOT	FastText	97.19 \pm 0.55	95.45 \pm 1.72	95.45 \pm 0.90
RF	Glove	97.10 \pm 0.91	94.73 \pm 1.98	99.63 \pm 0.49
C.-M.	7B	97.08 \pm 1.09	97.72 \pm 1.22	97.72 \pm 1.12
TPOT	Glove	97.05 \pm 1.13	96.05 \pm 0.96	98.10 \pm 2.30
RoB	large	96.97 \pm 1.05	97.96 \pm 0.98	96.01 \pm 1.24
MLP	Glove	96.68 \pm 0.99	97.48 \pm 0.70	95.88 \pm 1.31
RoB	Base	96.56 \pm 0.63	96.04 \pm 1.23	97.10 \pm 0.68
MLP	FastText	96.04 \pm 1.09	97.24 \pm 1.04	94.87 \pm 1.41
SVM	FastText	95.99 \pm 0.91	97.36 \pm 0.48	94.66 \pm 1.48

Fig. 8. Precision, recall and f1-score obtained on BLESS dataset.

We observe that all the algorithms achieve good results for each of the two datasets. However, on Bless dataset, all the algorithms obtain very good results (F-score between 95.99 and 97.81) while performances are more unequal on the EVALution dataset (F-score between 65.18 and 81.84).

Algo	Emb/Size	μ F1	μ R	μ P
Claire-M	7B	81.85 \pm 3.25	79.47 \pm 9.07	85.77 \pm 5.97
Mistral	7B	79.80 \pm 3.34	77.87 \pm 9.54	83.39 \pm 5.87
Llama-2	7B	80.62 \pm 3.91	79.63 \pm 7.87	82.22 \pm 2.53
RoB	large	81.15 \pm 2.5	84.73 \pm 2.69	78,06 \pm 4.52
RoB	Base	80.28 \pm 1.28	83.46 \pm 5.27	77.73 \pm 3.43
SVM	Glove	73.36 \pm 0.48	72.77 \pm 0.85	74.00 \pm 1.63
TPOT	FastText	72.70 \pm 0.91	72.66 \pm 1.18	72.73 \pm 0.71
TPOT	Glove	72.60 \pm 1.84	72.23 \pm 3.39	73.07 \pm 1.14
MLP	FastText	71.03 \pm 1.78	70.43 \pm 2.84	71.70 \pm 1.51
MLP	Glove	69.32 \pm 2.09	69.47 \pm 3.82	69,31 \pm 1.82
SVM	FastText	68.78 \pm 1.42	68.03 \pm 1.74	69,56 \pm 1.45
RF	Glove	65.70 \pm 1.30	64.04 \pm 1.80	67.47 \pm 1.47
RF	FastText	65.19 \pm 1.50	63.88 \pm 1.49	66.56 \pm 1.81

Fig. 9. Precision, recall and f1-score on EVALution dataset.

This performance difference likely stems from the EVALution corpus encompassing diverse relationship types, whereas we solely focused on 'is-a' relation-

ship. This limited our training data for experimentation and fine-tuning. On both datasets, the best results were obtained by LLMs: Llama-2 on BLESS and Claire-Mistral on EVALution, using 7B embedding models for both. We also tested Mistral for text generation. However, the results were quite similar to those achieved by Claire-Mistral and Mistral in classification tasks (on BLESS 97.23 versus 97.08 and 97.31 respectively). Even though variations remain limited, we observe an influence of embedding models on each algorithm.

While algorithms using pre-trained vector representations (RF, SVM, MLP) achieved excellent results on the Bless dataset, their performance dropped significantly on the second dataset. As previously discussed, the EVALution dataset may present greater processing challenges due to its inherent complexity. TPOT, RoBERTa achieve good results on the two datasets with slight variations depending on the chosen embedding.

Algo	Emb/Size	μ F1	μ R	μ P
CBert	Large	91.61 \pm 0.74	93.63 \pm 1.11	89.69 \pm 1.03
CBert	Base	90.97 \pm 0.96	93.35 \pm 2.18	88.75 \pm 1.14
C.-M.	7B	88.72 \pm 1.41	91.32 \pm 4.20	86.50 \pm 2.89
Llama-2	7B	88.17 \pm 1.54	89.35 \pm 2.48	87.28 \pm 4.65
TPOT	FastText	87.17 \pm 0.61	88.57 \pm 1.96	85.86 \pm 1.38
SVM	FastText	84.91 \pm 0.82	84.80 \pm 1.4	85.03 \pm 0.38
MLP	FastText	84.34 \pm 1.21	84.73 \pm 2.80	84.17 \pm 3.59
RF	FastText	80.53 \pm 1.57	76.87 \pm 2.34	84.60 \pm 0.91
TPOT	Glove	79.92 \pm 0.51	81.42 \pm 1.46	78.51 \pm 1.01
SVM	Glove	77.90 \pm 0.96	78.18 \pm 2.19	77.67 \pm 0.76
MLP	Glove	77.04 \pm 0.89	79.24 \pm 2.14	74.99 \pm 0.44
RF	Glove	74.25 \pm 1.88	70.84 \pm 2.56	78.03 \pm 1.12

Fig. 10. Precision, rappel and f1-score obtained on VOCAGEN dataset.

VOCAGEN dataset results Figures 10 present results obtained with each classification algorithm and each model on VOCAGEN dataset. The best results are obtained with Camembert (CBERT in table), slightly better for large embedding than for base embedding (respectively 91.61 and 90.97) with a very small standard deviation. Llama-2 and Claire-Mistral confirm the good results obtained on the literature datasets with very good scores as well. We observe that the scores of linear algorithms are slightly lower, as for the evaluation dataset, which confirms the difficulty of our task. There are slight differences depending on the embedding model used but the best performance is obtained with fastText.

A preliminary analysis was carried out on the best results obtained. These results showed different types of possible errors. For example, the terms “coiffeuse” (vanity table) and “meuble” (table) are an interesting error. In French, “coiffeuse” can refer to both a profession (hairdresser) and a specific type of furniture (vanity table). This ambiguity can explain why the model makes mistakes.

More generally, a small portion of the relationships contained in the dataset are not IS-A type relationships, but rather Part-of type relationships or others, for example “Europe” and “Eurasia” or “moldy” and “mushroom”. A cleaning step is underway to remove these relationships. Most classification errors stem from complex technical terms or terms that have different meanings in the context of construction. For example, “poutre longrine” translates to “longrine beam”, a specific type of beam used in construction. Similarly, “tuyau dauphin” is a French technical term for a type of flexible hose used for suction and discharge of water, although it literally translates to “dolphin hose”. The last category of errors involves brand mentions within relationship terms. For example, “Bic pen” or “Merlin axe”, here Merlin is a brand of an axe. In our context, the model needs to understand sentences like “put away the Merlins” and identify that the user is referring to axes.

6 Conclusion and future work

The paper reports our experiments and results for building a precise and large terminology for the construction domain. Collecting terminology is indeed the first step towards a complete knowledge model containing both concepts and relationships. During our work we were faced to several problems: finding resources and selecting them for building an appropriate corpus, thinking and developing pre-processing for cleaning those resources, experimenting distinct measures for n-grams and selecting the most appropriate, improving results by adding linguistic patterns and Internet queries. The current results are quite promising according to the evaluation of the extracted terminology carried out by 6 experts in the field. However, as manual evaluation of the produced terminology proved to be time-consuming and laborious, we developed in a second time a model to automatically or semi-automatically system to validate the terminology using our second partner’s knowledge model. Our goal was to propose a module for the detection of hypernyms performing and different combinations were tested for different corpora/languages. We explore two approaches for representing text data, the first uses classical words embedding vectors as features, and the second one relies on end-to-end approaches. Models fine-tuned with CamemBERT achieved a very promising F-score of 91.6%. While construction terminology may appear specialized and technical, deep learning approach with LLM and BERT model trained on a massive dataset seems to be sufficient for this task. From our French hypernymy relation model trained on VOCAGEN dataset and validated on the two other datasets, the next step will be to organize the extracted terminology to build a significantly richer taxonomy. This work validates the possibility of changing domains (a key objective for one of the partners) and maintaining good performance. We propose first improving results by merging the outputs of all algorithms, assuming they make errors on different data. A more in-depth analysis of the errors is necessary to validate this hypothesis. Another interesting perspective for this work would be to explore using the

knowledge model from our partner as input to enhance the model's prediction capabilities.

Acknowledgement

The authors would like to thank Mickaël Kessler for his help for reviewing and entering the results.

Bibliography

- Altman, E. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. In *The Journal of Finance*, 23(4). doi:10.2307/2978933, pp. 589–609.
- Amjadian, E., D. Inkpen, T. S. Paribakht, and F. Faez (2016). Local-global vectors to improve unigram terminology extraction. In *Proceedings of the 5th International Workshop on Computational Terminology*, pp. 2–11.
- Asim, M. N., M. Wasim, M. U. G. Khan, W. Mahmood, and H. M. Abbasi (2018, 10). A survey of ontology learning techniques and applications. *Database 2018*, bay101. bay101.
- Baroni, M., R. Bernardi, N.-Q. Do, and C. chieh Shan (2012). Entailment above the word level in distributional semantics. In *EACL*, pp. 23–32.
- Baroni, M. and A. Lenci (2010). Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics* 36(4), 673–721.
- Bestgen, Y. (2017). Évaluation de mesures d’association pour les bigrammes et les trigrammes au moyen du test exact de fisher (using fisher’s exact test to evaluate association measures for bigrams and trigrams). In *Actes des 24ème Conférence sur le Traitement Automatique des Langues Naturelles. Volume 2-Articles courts*, pp. 10–18.
- Biébow, B., S. Szulman, and A. J. B. Clément (1999). Terminae: A linguistics-based tool for the building of a domain ontology. In D. Fensel and R. Studer (Eds.), *Knowledge Acquisition, Modeling and Management*, pp. 49–66.
- Bojanowski, P., E. Grave, A. Joulin, and T. Mikolov (2017). Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics* 5, 135–146.
- Bouma, G. (2009). Normalized (pointwise) mutual information in collocation extraction. *Proceedings of GSCL 30*, 31–40.
- Brown, T., B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Nee-lakantan, P. Shyam, G. Sastry, A. Askell, et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems* 33, 1877–1901.
- Chowdhury, M. F. M., A. M. Gliozzo, and S. M. Trewin (2018, September 27). Domain-specific terminology extraction by boosting frequency metrics. US Patent App. 15/469,766.
- Cimiano, P. and J. Völker (2005). text2onto. In *International conference on application of natural language to information systems*, pp. 227–238. Springer.
- Cohen, J. (1960). A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* 20(1), 1–10.
- Cunningham, H. (2002). Gate, a general architecture for text engineering. In *Computers and the Humanities*, Volume 36, pp. 223–254.
- Daille, B. (2002, 12). Study and implementation of combined techniques for automatic extraction of terminology. *The Balancing Act: Combining Symbolic and Statistical Approaches to Language* 1, 1–8.

- Devlin, J., M.-W. Chang, K. Lee, and K. Toutanova (2019, June). BERT: Pre-training of deep bidirectional transformers for language understanding. In J. Burstein, C. Doran, and T. Solorio (Eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Minneapolis, Minnesota, pp. 4171–4186. Association for Computational Linguistics.
- Dias, G. and H.-J. Kaalep (2003). Automatic extraction of multiword units for estonian : Phrasal verbs. In *Languages in Development*, pp. 41:81–91.
- Drouin, P. (2003). Term extraction using non technical corpora as point of leverage. In n. Terminology, vol. 9 (Ed.), *John Benjamins Publishing Company: Amsterdam/Philadelphia*, pp. 99–115.
- Gillam, L., M. Tariq, and K. Ahmad (2005). Terminology and the construction of ontology. *TERMINOLOGY 11*, 55–81.
- Goutte, C. and E. Gaussier (2005). A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. *ECIR 2005 1*, 345–359.
- Harris, Z. S. (1968). *Mathematical Structures of Language*. New York: Wiley.
- Hearst, M. A. (1992). Automatic acquisition of hyponyms from large text corpora. In *COLING 1992*, pp. 1–7.
- Hunter, J., J. Louradour, V. Rennard, I. Harrando, G. Shang, and J.-P. Lorré (2023). The claire french dialogue dataset. *arXiv preprint arXiv:2311.16840 1*, 1–11.
- Kessler, R., N. Béchet, and G. Berio (2019). Extraction of terminology in the field of construction. In *Proceedings of DDP 2019*, pp. 22–26.
- Kluegl, P., M. Toepfer, P.-D. Beck, G. Fette, and F. Puppe (2016). Uima ruta: Rapid development of rule-based information extraction applications. *Natural Language Engineering 22*(1), 1–40.
- Lang, C., R. Schneider, and K. Suchowolec (2018). Extracting specialized terminology from linguistic corpora. *GRAMMAR AND CORPORA 1*, 425.
- Lefever, E., L. Macken, and V. Hoste (2009). Language-independent bilingual terminology extraction from a multilingual parallel corpus. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics, EACL '09*, Stroudsburg, PA, USA, pp. 496–504. Association for Computational Linguistics.
- Lenci, A. and G. Benotto (2012, 7-8 June). Identifying hypernyms in distributional semantic spaces. In **Proceedings of (SemEval 2012)*, Montréal, Canada, pp. 75–79. Association for Computational Linguistics.
- Levy, O., S. Remus, C. Biemann, and I. Dagan (2015, May–June). Do supervised distributional methods really learn lexical inference relations? In *Proceedings of NACL 2015*, Denver, Colorado, pp. 970–976.
- Lin, D. (1998). An information-theoretic definition of similarity. In *Proceedings of ICML '98*, San Francisco, CA, USA, pp. 296–304. Morgan Kaufmann Publishers Inc.
- Liu, H. and P. Singh (2004, October). Conceptnet — a practical commonsense reasoning tool-kit. *BT Technology Journal 22*(4), 211–226.

- Liu, Y., M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov (2019). Roberta: A robustly optimized bert pretraining approach.
- Luu, A. T., Y. Tay, S. C. Hui, and S. K. Ng (2016, November). Learning term embeddings for taxonomic relation identification using dynamic weighting neural network. In *Proceedings of the 2016 Conference on EMNLP*, Austin, Texas, pp. 403–413.
- Ma, N., D. Wang, H. Bao, L. He, and S. Zheng (2023, December). KEPL: Knowledge enhanced prompt learning for Chinese hypernym-hyponym extraction. In H. Bouamor, J. Pino, and K. Bali (Eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, Singapore, pp. 5858–5867. Association for Computational Linguistics.
- Macken, L., E. Lefever, and V. Hoste (2013). Taxis: bilingual terminology extraction from parallel corpora using chunk-based alignment. *Terminology* 19(1), 1–30.
- Markchom, T., H. Liang, and J. Chen (2022, July). UoR-NCL at SemEval-2022 task 3: Fine-tuning the BERT-based models for validating taxonomic relations. In G. Emerson, N. Schluter, G. Stanovsky, R. Kumar, A. Palmer, N. Schneider, S. Singh, and S. Ratan (Eds.), *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, Seattle, United States, pp. 260–265. Association for Computational Linguistics.
- Martin, L., B. Muller, P. J. Ortiz Suárez, Y. Dupont, L. Romary, É. de la Clergerie, D. Seddah, and B. Sagot (2020, July). CamemBERT: a tasty French language model. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, Online, pp. 7203–7219. Association for Computational Linguistics.
- McHugh, M. L. (2012). Interrater reliability: the kappa statistic. In *Biochemia medica*, pp. 1–7.
- Meignier, S. and T. Merlin (2010). Lium spkdiarization: an open source toolkit for diarization. In *in CMU SPUD Workshop*.
- Meyers, A. L., Y. He, Z. Glass, J. Ortega, S. Liao, A. Grieve-Smith, R. Grishman, and O. Babko-Malaya (2018). The termolator: Terminology recognition based on chunking, statistical and search-based scores. *Frontiers in Research Metrics and Analytics* 3, 1–14.
- Mikolov, T., E. Grave, P. Bojanowski, C. Puhersch, and A. Joulin (2018). Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, pp. 1–4.
- Mikolov, T., I. Sutskever, K. Chen, G. Corrado, and J. Dean (2013). Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS'13*, USA, pp. 3111–3119. Curran Associates Inc.
- Mirkin, S., I. Dagan, and M. Geffet (2006, July). Integrating pattern-based and distributional similarity methods for lexical entailment acquisition. In *Proceedings of the COLING/ACL 2006*, Sydney, Australia, pp. 579–586. ACL.
- Panchenko, A., S. Faralli, E. Ruppert, S. Remus, H. Naets, C. Fairon, S. P. Ponzetto, and C. Biemann (2016). TAXI at semeval-2016 task 13: a taxonomy

- induction method based on lexico-syntactic patterns, substrings and focused crawling. In *Proceedings of the 10th International Workshop on Semantic Evaluation, SemEval@NAACL-HLT 2016, San Diego, CA, USA, June 16-17, 2016*, pp. 1320–1327.
- Pauwels, P. and W. Terkaj (2016, 03). Express to owl for construction industry: Towards a recommendable and usable ifcowl ontology. *Automation in Construction* 63, 100–133.
- Pennington, J., R. Socher, and C. D. Manning (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.
- Roche, M. and Y. Kodratoff (2004). Exit: Un système itératif pour l’extraction de la terminologie du domaine à partir de corpus spécialisés. In *Proceedings of JADT 4*, pp. 946–956.
- Roller, S., K. Erk, and G. Boleda (2014, August). Inclusive yet selective: Supervised distributional hypernymy detection. In *Proceedings of COLING 2014, Dublin, Ireland*, pp. 1025–1036. Dublin City University and ACL.
- Salton, G. and M. J. McGill (1986). *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc.
- Santus, E., F. Yung, A. Lenci, and C.-R. Huang (2015, July). EVALution 1.0: an evolving semantic dataset for training and evaluation of distributional semantic models. In *Proceedings of the 4th Workshop on LDL*, pp. 64–69.
- Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees.
- Shwartz, V., E. Santus, and D. Schlechtweg (2016). Hypernyms under siege: Linguistically-motivated artillery for hypernymy detection. *CoRR abs/1612.04460*, 425–435.
- Touvron, H., L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale, et al. (2023). Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* 1, 510–521.
- Vetter, K., M. Segiet, and K. Lennermann (2022, July). KaMiKla at SemEval-2022 task 3: ALBERTo, BERT, and CamemBERT—Be(r)tween taxonomy detection and prediction. In G. Emerson, N. Schluter, G. Stanovsky, R. Kumar, A. Palmer, N. Schneider, S. Singh, and S. Ratan (Eds.), *Proceedings of SemEval-2022*, Seattle, United States, pp. 282–290. ACL.
- Weeds, J., D. Clarke, J. Reffin, D. Weir, and B. Keller (2014, August). Learning to distinguish hypernyms and co-hyponyms. In *Proceedings of COLING 2014, Dublin, Ireland*, pp. 2249–2259.
- Weeds, J., D. Weir, and D. McCarthy (2004, aug 23–aug 27). Characterising measures of lexical distributional similarity. In *Proceedings of COLING 2004*, pp. 1015–1021.
- Wohlgenannt, G. and F. Minic (2016). Using word2vec to build a simple ontology learning system. In *International Semantic Web Conference (Posters & Demos)*.