Brand-Product Relation Extraction Using Heterogeneous Vector Space Representations

Arkadiusz Janz[◇], **Łukasz Kopociński**[◇], **Maciej Piasecki**[◇], **Agnieszka Pluwak**[♠]
[◇] Wrocław University of Science and technology
Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{arkadiusz.janz|lukasz.kopocinski|maciej.piasecki}@pwr.edu.pl

[♠] SentiOne Sp. Z O.O.
Lęborska 3B, 80-386 Gdańsk, Poland
agnieszka.pluwak@senti1.com

Abstract

Relation Extraction is a fundamental NLP task. In this paper we investigate the impact of underlying text representation on the performance of neural classification models in the task of Brand-Product relation extraction. We also present the methodology of preparing annotated textual corpora for this task and we provide valuable insight into the properties of Brand-Product relations existing in textual corpora. The problem is approached from a practical angle of applications Relation Extraction in facilitating commercial Internet monitoring.

Keywords: relation extraction, vector space models, deep learning

1. Introduction

Internet monitoring tools, developed and commonly used for business purposes, heavily depend in their internal implementation on search engines and keyword or key-phrase based search queries. Commonly such queries are built manually either by the operators of the monitoring systems or in cooperation with the final users. The ability of the monitoring system to generate suggestions in a reactive way is a basic requirement of user experience. Such a system should propose query expansion (or more complex adaptation) by morpho-syntactically related words (inflected forms), semantically interrelated words (synonyms, hypernyms, meronyms, but also derivations), etc., and also words connected in a domain specific way (named entities and their attributes or pairs of named-entities related in different ways). In the case of inflection morphological dictionaries can be successfully applied. Semantic relations can be extracted from lexical databases like WordNet[1] or acquired by distributional semantics means. However, discovery of relations between named-entities is a complex issue, combining named-entity recognition on one hand and relation detection on the other. In this paper we focus on the problem of discovery and acquisition of named-entity pairs related by the *brand* (a producer) – *product* relation as a data source for the query expansion mechanism in an internet monitoring system. Monitoring actions and search queries for them can be related to specific business sectors or product types, e.g. cosmetics, electronic devices, specific brands (Samsung, Nivea, Apple, H&M etc.), or specific products (Lays chips, iPhone, Head&Shoulders shampoo). In all such cases, the extraction of relation between a product and a brand is crucial from the point of view of the search system or suggestions for the user building the query. In the case of a business sector query, this relation helps finding products related to brands from this sector and so enriching the search results with new data. In the case of a brand query, this relation also helps finding, e.g., opinions not only about the brand, but also its products. In the case of product search, opinions not only about products, but also about their brands can be found if necessary. Finally, in all three types of queries, the relation facilitates defining search profiles for competing brands and their products.

2. Related Work

The problem of relation extraction is one of the most important tasks in the area of Information Extraction and has a very long history in general NLP research. A classic example of relation extraction presented by (Hearst, 1992) concerned the recognition of hyponymy relation in unstructured textual corpora by applying predefined lexico-syntactic patterns on the text. Pattern-based approaches were partially replaced later with more automated methods being capable of learning the patterns directly from the text (Snow, 2004) (given a small annotated seed of already known word pairs linked by hyponymy relation). However, such approaches are limited to relations expressed by relatively stable and specific patterns and may result in good precision but mostly low recall, that is not acceptable for the construction of monitoring search queries.

With the progress of neural representation learning (word embeddings, contextual embeddings) and neural classification methods (bidirectional recurrent models, convolutional networks, and finally transformers) new approaches have emerged e.g. (Shwartz et al., 2016; Nguyen et al., 2017; Roller et al., 2018; Qu et al., 2018; Eberts and Ulges, 2019). They improved generalisation of the relation models by mapping context onto vector spaces.

New contextualised text representations such as deep bidirectional language models e.g. ELMo(Peters et al., 2018), transformer networks e.g. BERT(Devlin et al., 2018), RoBERTa(Liu et al., 2019), TransformerXL(Dai et al., 2019) have been proposed to significantly improve the performance across a range of natural language processing tasks.

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

The solutions to the task of relation extraction for named-entities usually follow the pipeline approach where the subtasks of entity detection and relation recognition are solved separately. These kind of approaches train two separate models and combine their results to extract semantic relations linking entities in text. However, in the case of a pipeline approach different components can be trained on different datasets. As datasets are often created in different projects, a situation in which there are separate training-testing sets, e.g. for named-entities (a large, rich one) and relations (much smaller and focused on representing relations) is quite frequent.

The tasks of entity detection and relation extraction may benefit from each other when they are combined within the same model and solved together (Figure 3). Recently the joint approaches have become more popular also in other NLP areas. (Luan et al., 2019) has introduced a general framework for relation extraction that is capable of detecting the entities, resolving coreferences and identifying relations between detected entities. The framework was based on the idea of multitask learning paradigm where the loss function was a combination of the log-likelihood values of entity recognition task, relation recognition and coreference resolution. (Eberts and Ulges, 2019) has proposed a joint model for entity identification and relation extraction based on BERT transformer. The architecture of the model consisted of three subtask-oriented layers: i) span classification, ii) span filtering (to detect entities occurring in the text), and iii) relation classification. Some of these approaches are strongly based on the implicit assumptions that a relatively large joint training set combing named-entity annotation and relation annotation in one place is available, and a pre-trained complex sequential language model is available, as well. These pre-conditions are not always fulfilled, and that is why we search for a less restrictive approach.

3. Overview

3.1. Task Definition

The general task of relation extraction is to extract from unstructured textual corpora all the triples (e_s, r, e_t) consisting of relevant entities e_s, e_t and a relation r linking these entities. We define our task as a special case of relation extraction problem. The task requires to recognise entities representing *brands* and their *products* appearing in the text and decide whether they are semantically connected by *brand-product* relation (which means the *product* was probably manufactured by the company owning given *brand*). The task can be approached in a combined way, but also in a component-based way i.e. by applying a separated named-entity recognition component and combining its work with the relation recognition.

The problem is illustrated by the two examples (translated from Polish) below that immediately reveal one characteristic problem: a brand name and a product name often occur consecutively without any space in between, so without any linking pattern.

*"The **[Brand-A]**^{e1} **[Product-X]**^{e2} has an impressive camera, but the battery life is too short. **[Product-X]**^{e2} has been redesigned a bit, so it's slightly different than its predecessor **[Product-Y]**^{e4}."*

*"**[Brand-B]**^{e1} **[Product-Z]**^{e2} is fantastic! It keeps your skin feeling soft and really smooth. Using **[Brand-B]**^{e1} **[Product-Z]**^{e2} regularly will definitely..."*

3.2. Challenges

Despite the fact that supervised neural models mostly express very good performance in NER and relation recognition, they still suffer from the lack of annotated data. Data scarcity is a major challenge in the task of relation extraction. A large and representative collection of samples with expensive gold annotations must be acquired that may be challenging for specific domains. The problem of extracting relations between entities is even more difficult due to the fact that there are usually many different groups of domain-specific entities that should be covered by the model (e.g. brands representing *clothes*, *record labels*, *cosmetics*, *cars* etc.). For these reasons, it is very difficult to develop a solution that is robust and effective from a practical point of view, i.e. from the point of view of commercial applications.

As our task is directly related to named-entity recognition (NER) first we studied the properties and distribution of NEs representing *brands* and *products* in the annotated corpora from the three domains used during experiments. Because the intended model is supposed to recognise a pair of NEs as linked by the brand-product relation on the basis of the contextual properties, we analysed also the relative distribution of *brands* and *products* in the corpora, especially the distances of them in co-occurrences. The data are presented in Figure 1. The average distance between source and target entities in the text (Figure 1) is close to zero (no spacing between *brand* tokens and *product* tokens), which means the brands and the products are usually joined together in the text. This issue makes our problem very difficult to solve, both for *joint* relation extraction approaches and *pipeline* approaches because they both rely on NER. The typical NER model joins the names occurring together in the text and treats them as a single named-entity. This means also that the models are strongly affected by the issue of *lexical memorization* (Levy et al., 2015) that limits their generalization capability (see Section 6.), because the model will try to memorize if the specific names should be treated separately or not. To prevent this issue it is necessary to include specific world knowledge about existing *brands* and their potential *products*.

In this paper we would like to evaluate modern vector space representations in the task of relation extraction and to check their generalization capabilities of proposed models in a practical setting for three selected domains: *Phones*, *Banking*, and *Cosmetics*.

3.3. Contributions

To summarise, our main contributions are as follows:

- We present a study of relation extraction process in the practical setting pointing to potential problems that new approaches emerging in this area should be able to resolve in the future.

- We present a model designed for limited data e.g. lack of annotated data for specific domain, but also lack of annotated data for specific subtask (small annotated data for NER when the task is solved in a joint way with relation extraction).

- We propose a model combining different vector space representations of varied complexity and suitable for relatively easy tuning to a given domain.

4. Dataset

Most NER solutions are developed and evaluated on datasets in which traditional NE categories, such as first and last name, geographic names or institutions, dominate. However, in the case of commercial internet monitoring, the main focus is given to brands, their related products and versions. As such NEs are less represented in existing Polish corpora, a new corpus had to be collected and annotated for the purpose of this task. In order to achieve broad coverage, wide selection and good quality of data, we used the *SentiOne Listen* tool[2] – an internet monitor tool developed for business purposes. SentiOne Listen provides a panel for building rules of internet monitoring, i.e. implemented as a complex search queries. The panel itself enables the selection of relevant results of the performed search queries and delivers necessary information about the metadata of texts, such as source, access or time of retrieval. During 2018 the SentiOne panel was applied to collect altogether 10,346,021,032 mentions among which 739,278,408 were written in the Polish language. The search process was controlled by a team of media monitoring analysts. The crawling search rules were built around proper names as seed keywords and next expanded for the sake of targeting precision.

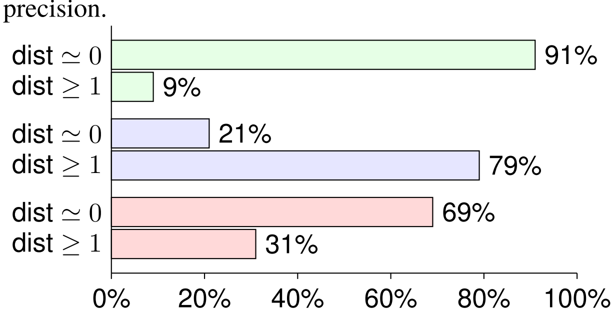

Figure 1: Distance distribution between *brand* and *product* in the corpus: green – *Phones*, violet – *Banking*, red – *Cosmetics*.

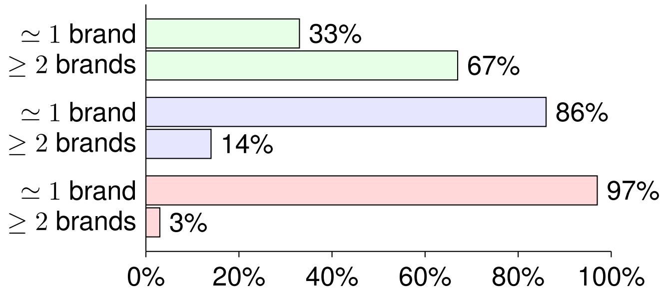

Figure 2: The number of texts with only one brand (being linked with some product) vs the texts having more than one brand (also being in relation with some product). Green – *Phones*, violet – *Banking*, red – *Cosmetics*.

[2] `https://sentione.com/pl/rozwiazania/narzedzie-do-social-listeningu`

The dataset quality was further improved by taking into account metadata of the sources and guiding the search process towards higher probability of occurrence of opinionated texts, such as microblogs, blogs or forums. Mostly an occurrence of a single keyword from a list of proper names was considered to be sufficient for a given text to be accepted, with the exception of the domains of *banking* and *cosmetics*, where it had to be accompanied by domain-related terms (domain specific heuristics). All the keywords in every query were manually inflected with regular expressions to provide better coverage of the search (limitations of a standard search engine).

The corpus collection process was further tuned in its scope in the second stage by limiting the query results to relevant word senses only. Such a restriction refers to word senses and metadata and were imposed manually due to the limited accuracy and efficiency of Word Sense Disambiguation tools. After the manual analysis of the search results an analyst could decide about excluding some irrelevant meanings resulting from, for instance, the homography of proper names vs common words, e.g. *lovely* (adjective and brand name). Topics related to frequently repeating mentions (a statistical peak of mentions) were also restricted in a query in order to avoid repetitions of the same topics, possibly dominating the corpus, such as e.g. high numbers of comments to some company's image crisis. Finally, further keyword restrictions were added, such as e.g. on account or foundation in order to exclude searches related with charity actions, which belong to statistical mention peaks, too. Finally, a few restrictions of the crawling metadata (e.g. source or domain, authors, comments versus articles) were applied as well. To provide some examples, texts authored by brands in the social media were excluded. In addition, search results from some specific sources or even specific websites that had been recognised on some stage as introducing irrelevant texts were removed from the list. As a result, a rich and diversified sample texts from different sources such as user forums, websites with internet opinions, portals of internet shops, blogs, popular media sites, as well as social media were collected.

4.1. Corpus Annotation

A large sample of texts was randomly selected from all domain corpora and manually annotated as a training-testing dataset. Both proper nominal named-entities were annotated, their derived adjectives and related numbers, e.g. *Wedel* (a well-known Polish chocolate manufacturer) as a proper name and *wedlowski* as an adjective derived from it, *S8* as a version of the *Samsung* (a brand name) smartphone. On the basis of the initial linguistic analysis, the following tagset was assumed:

- BRAND NAME - the name of a company like e.g. '*Samsung*', '*Coca-Cola*', '*Nivea*',
- PRODUCT NAME - the name of a product or model like e.g. '*Galaxy*', '*Astra*', '*Sensitive*',
- VERSION - version of a specific product, typically in the domain of electronic devices, e.g. '*S8*', '*P10*', '*2.0*',
- BRAND NAME IMP - version or product names that

stand for or imply a brand name, for example "*Mam S8*" (*I have an S8*), that implies someone has a *Samsung Galaxy* smartphone,

- PRODUCT NAME IMP - brand names that stand for products, "*Posmarowałam ręce Dove*" (*I have moisturized my hands with Dove*),
- PRODUCT ADJ - the adjective derived from product's name, like '*galaksowy*' (derived from *Galaxy*), '*siódemkowy*' (derived from *Windows 7*),
- BRAND ADJ - the adjective derived from a company's name, '*wedlowski*' (produced by *Wedel*), '*samsungowy*' (produced by *Samsung*), '*applowski*' (derived from *Apple*), '*rossmanowy*' (related with *Rossman*).

The proposed tagset was used to provide a more accurate description for possible brand and product entities. To capture entities that can change their morphosyntactic categories we used BRAND_ADJ and PRODUCT_ADJ annotations (*Samsung* as a *noun*, or *samsungowy* representing *Samsung* as an *adjective*).

Domain	#B-P relations	#B entities	#P entities
Phones	4 261	9 049	7 596
Banking	216	4 694	899
Cosmetics	1 893	10 896	3 213

Table 1: The overall distribution of annotations: the number of *brand-product* relations (#B-P rels.), the number of annotated *brand* entities (#B-ent.), and the number of annotated *product* entities (#P-ent.).

A collection of 18,386 texts in total was selected for the annotation process. A team of 5 trained linguists tagged the corpus after the inter-annotator agreement of kappa > 0.8 had been achieved on the subset of 300 texts. The total number of all NER tags in the whole corpus after annotation is 65,611 and the total number of annotated texts (with at least one NER tag) is 13,694. These numbers can be further characterised more precisely with respect to the domains.

4.2. Problematic cases in annotation

Since the corpus is a collection of the computer-mediated texts, characterized with an informal style, several problems were noticed during the tagging process. First, some proper names were impacted by typos and down-cased. They were tagged regularly as uppercase proper names, but with an attribute adding their correct formal name (e.g. *samsung* $\rightarrow$ *Samsung*). This is even more problematic if a down-cased named-entity becomes a regular lexeme, e.g. *Zielona Oliwka* ('Green Olive') is a type of a cosmetic, but if down-cased it means just an olive oil of green colour. Another characteristic feature of this corpus and the Polish language was inflection of proper names, and different, colloquial spelling variants of the proper names or foreign proper names becoming loan words and receiving the Polish spelling and inflection. Therefore, we have observed that the popular cosmetics brand *Nivea* could be spelled e.g. *Niwea* and inflected as *Niwei*, *Niweą*, etc. in different grammatical cases. In such cases, both the lemma and the correct proper name were provided. Especially proper names difficult to spell like e.g. 'Buorjois' are rich in creative new variants e.g. *Burjous* or even *Burżyły*. Another interesting challenge of the corpus annotation were elliptical constructions and metonymies which have encouraged us to introduce tags for implied contents, e.g. a producer name used as a product name.

Our analysis shows that specific features and shorthands are characteristic for colloquial style. Thus, a lot of information about products and brands is lost, if we consider only explicitly provided data. An introduction of implicit tags solved the problem transferring the implied human knowledge to the system for enriched reasoning and data extraction without losses.

5. Methods

To recognize *brand-product* relations we use a neural model with pre-trained ELMo (Peters et al., 2018) embeddings as a baseline representation that we use to model source and target entities (s, t) in the text. An input text is tokenized and represented as a sequence of tokens. In case of multiword entities we treat the entities as a span of tokens and use their vector space representations to generate the final representation of the entities. Let $(s_1, s_2, ..., s_n)$ denote a span representing all tokens of our source entity s, and $(t_1, t_2, ..., t_m)$ the tokens of our target entity t. The final representation of a given multiword entity is obtained by applying max pooling function on its vectorised elements $e(s) = f_{maxpool}(v(s_1), v(s_2), ..., v(s_n))$, and $e(t) = f_{maxpool}(v(t_1), v(t_2), ..., v(t_m))$. In our first experimental setting we assume that s and t entities (and their spans) are already marked in the text since we are interested more in the general ability of the model to recognize relations with respect to the domain. In our second experimental setting we use a pre-trained NER model to detect the entities with their spans first (pipeline approach), but instead of directly relying on NER annotations, we use them as additional features for the model. Two binary values, for source and target respectively, are added to the features vector. For spans where each token was marked by NER as named-entity, value of 1 is assigned. For those in which at least one token is not named-entity 0 is assigned.

Our method is based on a heterogeneous vector space model used to represent entities in the context. The baseline representation (ELMo) is expanded by incorporating complementary vector space representations to inject more knowledge into the model. In this paper we explore the impact of complementary vector space models and simple linguistic features on the performance in the task of *brand-product* relation extraction. While the recent work in the area of relation extraction is focused more on utilising transformer networks (e.g. BERT transformer), we decided to adopt deep contextualized ELMo embeddings. Since the BERT transformer was trained on the task of next sentence prediction it provides some additional knowledge about potential relations between sentences in the context. This property gives BERT some advantage over ELMo due to the fact that ELMo embeddings are limited to the scope of single sentences. However, because of its complexity a pre-trained BERT transformer published by Google is difficult to retrain on new corpora.

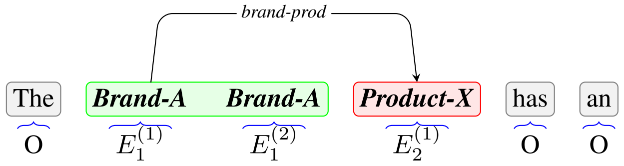

Figure 3: Brand-Product relation extraction seen as a joint task with named entity recognition and relation classification.

5.1. Vector Space Representations

As it was stressed in Section 5. our baseline vector space representation uses ELMo embeddings to model source and target entities (s, t). The baseline model is expanded with context insensitive fastText (Joulin et al., 2016) embeddings of source and target entities to capture general semantic meaning of analysed entities without taking into account their context-dependent properties. We also provide an additional vector representation $e(c)$ (one for a source entity, and one for a target) of extended context c which is a small textual window of few sentences (a sentence with the entity, one sentence before, and one sentence after).
The vector representation of our extended context c is obtained by applying sent2vec (Pagliardini et al., 2017) model on the textual window with a simple masking procedure excluding source and target entities from this window. This representation completes the model with contextual information (e.g. important domain vocabulary and other domain-specific entities existing in the context) that is not directly available in ELMo embeddings.
Let $(m_1, m_2, ..., m_p)$ denote all vector space models (e.g. ELMo, fastText, etc.) used to generate a sequence of vector space representations $(e_{m_1}(s), e_{m_2}(s), ..., e_{m_p}(s))$ for the source and $(e_{m_1}(t), e_{m_2}(t), ..., e_{m_p}(t))$ for the target entities with respect to the model being applied. The final embeddings (x_s, x_t) of our source and target entities (s, t) are computed by applying concatenation operator on their vector sequences:

$$x_s = e_{m_1}(s) \oplus e_{m_2}(s) \oplus ... \oplus e_{m_p}(s) \tag{1}$$

$$x_t = e_{m_1}(t) \oplus e_{m_2}(t) \oplus ... \oplus e_{m_p}(t) \tag{2}$$

In case of sent2vec representation, we decided to provide only one vector for both of the entities to avoid data redundancy. The vector space representations of source and target entities should be the same because they occur in the same context. Thus, the final vector representation treated as an input of our neural model is as follows:

$$x = x_s \oplus e_{s2v}(c) \oplus x_t \tag{3}$$

5.2. Relation Recognition

We treat our problem as a binary classification task. Given a tuple of training entities (s, t) and their class label r (*in-relation* vs *no-relation*) we train a supervised model which is the Feed Forward neural network with two densely connected hidden layers, dropout, and single classification layer on top (see Figure 4).

6. Evaluation Procedure

To measure the performance of our approaches we adopt two major evaluation strategies: splitting the data *randomly* (and test on held out data), or splitting the data *lexically* as it was suggested in (Levy et al., 2015) (*lexical split*). The latter strategy requires that our train examples and test examples have no lexical overlap, which means the entities are completely different in both of the subsets. The *lexical split* strategy guarantees that the final error estimate of our model should be more reliable since we avoid the problem of lexical bias towards our training samples (in this case, the input entities s and t). The bias is directly connected with a specific property of supervised models called *Lexical Memorization* (Levy et al., 2015). We tested our models on three subdomains: *Phones*, *Banking*, *Cosmetics* (see Section 4.).

7. Results

In our first experimental setting we tested classifier trained on the dataset with all of the domains mixed together. The model performs quite well if we take the *random split* approach, which means the same entities can repeat in the training data as well as in the test data, but they can appear in different contexts. The same trend can be observed both in mixed domain setting (Table 2) and single domain setting (Table 4). Our second experimental part was focused on *lexical split* approach. It shows that the performance of the model significantly drops when we test our model on new unseen entities (Tables 2 and 5). The results obtained in our second experimental setting suggest that the model is focused more on lexical properties directly connected with word forms of analysed entities rather than their contextual properties. It also suggests that it is not clear how well a pretrained vector space representation reflects lexical, syntactic, and semantic properties of given entities and their contexts.

Repr.	F_+	F_-	$F_+^{(ls)}$	$F_-^{(ls)}$
M1: ELMo	0.937	0.967	0.393	0.845
M2: ELMo + fT	0.940	0.970	0.406	0.850
M3: ELMo + s2v	0.932	0.965	0.387	0.848
M4: ELMo + fT + s2v	0.938	0.968	0.435	0.855

Table 2: The performance of the model for Mixed-Domain setting with respect to the underlying text representation. We provide F1-scores for *positive* class (entities are *in relation*) and *negative* class (there is *no relation* between entities). We also provide the results for *lexical split* evaluation.

The simple binary NER features (Section 5.) appeared to be quite effective and we observed yet another improvement for random split setting as well as for lexical split setting (Tables 3 and 6).
Our baseline representation was outperformed by our first expansion. The fastText embeddings of source and target entites improved the performance for both random split and lexical split approaches (see Table 2). Still, the model shows quite low performance in the lexical split setting. The best results were obtained by adding all expansions together. We added an additional sent2vec embedding of a context to ELMo and fastText embeddings of given entities. The

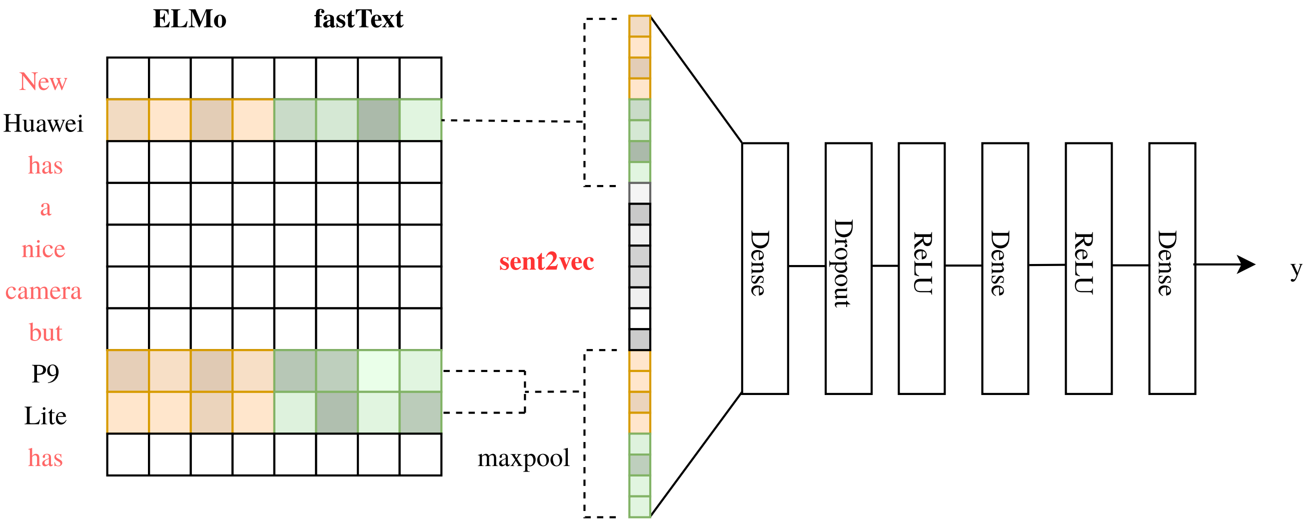

Figure 4: The neural architecture for relation extraction. The model combines contextual ELMo embeddings with context insensitive fastText embeddings, and the general sent2vec embedding of our context words (*red* words only).

Repr.	F_+	F_-	$F_+^{(ls)}$	$F_-^{(ls)}$
M1: ELMo	0.942	0.970	0.405	0.851
M2: ELMo + fT	0.946	0.972	0.419	0.853
M3: ELMo + s2v	0.938	0.968	0.423	0.853
M4: ELMo + fT + s2v	0.943	0.970	0.453	0.857

Table 3: The performance of the model for Mixed-Domain setting with respect to the underlying text representation containing NER features.

difference between single vector space representation and concatenated vector space representation seemed to be insignificant when we analysed the model in random split setting, but for lexical split setting the difference was more clear and a concatenated representation outperformed other approaches.

We also present the results for Single-Domain setting (Table 4), where the model is trained and tested for single domain only. In this setting, we noticed that our model follows the same trend as before, i.e. there is a significant performance drop when the model is tested for the *lexical split* setting (Table 5).

Model	Phones		Banking		Cosmetics	
	F_+	F_-	F_+	F_-	F_+	F_-
M1	0.940	0.970	0.881	0.937	0.935	0.965
M2	0.940	0.970	0.860	0.930	0.938	0.966
M3	0.941	0.970	0.894	0.945	0.933	0.964
M4	0.942	0.971	0.898	0.947	0.933	0.964

Table 4: The performance of the model for Single-Domain setting with respect to the underlying text representation (*random split*).

8. Conclusions

In this paper we attempted to propose a solution for a specific subtype of relation extraction problems i.e. the task of extracting relations between named-entities representing *brands* and their potential *products*. We presented a general methodology to prepare the annotated textual corpora for this task and we showed a statistical

Model	Phones		Banking		Cosmetics	
	$F_+^{(ls)}$	$F_-^{(ls)}$	$F_+^{(ls)}$	$F_-^{(ls)}$	$F_+^{(ls)}$	$F_-^{(ls)}$
M1	0.216	0.841	0.610	0.800	0.767	0.870
M2	0.156	0.837	0.576	0.799	0.770	0.871
M3	0.218	0.842	0.570	0.794	0.758	0.867
M4	0.179	0.838	0.626	0.809	0.760	0.867

Table 5: The performance of the model for Single-Domain setting with *lexical split* evaluation.

Model	Phones		Banking		Cosmetics	
	F_+	F_-	F_+	F_-	F_+	F_-
M1	0.949	0.974	0.874	0.932	0.936	0.965
M2	0.948	0.974	0.860	0.927	0.940	0.968
M3	0.951	0.975	0.897	0.945	0.934	0.964
M4	0.948	0.974	0.869	0.934	0.935	0.965

Table 6: The performance of the model for Single-Domain setting with respect to the underlying text representation containing NER features.

Model	Phones		Banking		Cosmetics	
	$F_+^{(ls)}$	$F_-^{(ls)}$	$F_+^{(ls)}$	$F_-^{(ls)}$	$F_+^{(ls)}$	$F_-^{(ls)}$
M1	0.218	0.842	0.559	0.794	0.768	0.871
M2	0.190	0.839	0.606	0.803	0.778	0.874
M3	0.258	0.845	0.629	0.803	0.765	0.870
M4	0.231	0.843	0.617	0.809	0.768	0.871

Table 7: The performance of the model for Single-Domain setting with *lexical split* evaluation containing NER features.

analysis of its properties. We also proposed a model for recognizing *brand-product* relations in unstructured textual corpora. Our evaluation study shows that the supervised models based on existing vector space representations do not scale well when they are used to extract relations between named-entities. The lexical split evaluation methodology shows that there is a great need of knowledge-aware vector space models for these kind of tasks. The code of our relation extraction tool and the distributional models used in this paper are freely available at `https://gitlab.clarin-pl.eu`[3]. We also make our corpus publicly available at `https://clarin-pl.eu/dspace`[4].

9. Acknowledgements

Project "SentiCognitiveServices - next generation service for automating voice of customer and social media support based on artificial intelligence methods" is co-financed by European Union through European Regional Development Fund under the Smart Growth Programme 2014 - 2020.

10. Bibliographical References

Dai, Z., Yang, Z., Yang, Y., Cohen, W. W., Carbonell, J., Le, Q. V., and Salakhutdinov, R. (2019). Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Eberts, M. and Ulges, A. (2019). Span-based joint entity and relation extraction with transformer pre-training.

Joulin, A., Grave, E., Bojanowski, P., and Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Levy, O., Remus, S., Biemann, C., and Dagan, I. (2015). Do supervised distributional methods really learn lexical inference relations? In Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 970–976.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Luan, Y., Wadden, D., He, L., Shah, A., Ostendorf, M., and Hajishirzi, H. (2019). A general framework for information extraction using dynamic span graphs. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, volume 1, pages 3036–3046.

Nguyen, K. A., Walde, S. S. i., and Vu, N. T. (2017). Distinguishing antonyms and synonyms in a pattern-based neural network. *arXiv preprint arXiv:1701.02962*.

Pagliardini, M., Gupta, P., and Jaggi, M. (2017). Unsupervised learning of sentence embeddings using compositional n-gram features. *arXiv preprint arXiv:1703.02507*.

Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*.

Qu, M., Ren, X., Zhang, Y., and Han, J. (2018). Weakly-supervised relation extraction by pattern-enhanced embedding learning. In Proceedings of the 2018 World Wide Web Conference, pages 1257–1266. International World Wide Web Conferences Steering Committee.

Roller, S., Kiela, D., and Nickel, M. (2018). Hearst patterns revisited: Automatic hypernym detection from large text corpora. *arXiv preprint arXiv:1806.03191*.

Shwartz, V., Goldberg, Y., and Dagan, I. (2016). Improving hypernymy detection with an integrated path-based and distributional method. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2389–2398.

11. Language Resource References

[3] https://gitlab.clarin-pl.eu/team-semantics/semrel-extraction

[4] https://clarin-pl.eu/dspace/handle/11321/736