Concept Extraction Using Pointer–Generator Networks and Distant Supervision for Data Augmentation

Alexander Shvets[1] [0000–0002–8370–2109] and Leo Wanner[1,2] [0000–0002–9446–3748]

[1] NLP Group, Pompeu Fabra University, Roc Boronat, 138, Barcelona, Spain
{alexander.shvets,leo.wanner}@upf.edu
[2] Catalan Institute for Research and Advanced Studies (ICREA)

Abstract. Concept extraction is crucial for a number of downstream applications. However, surprisingly enough, straightforward single token/nominal chunk–concept alignment or dictionary lookup techniques such as DBpedia Spotlight still prevail. We propose a generic open domain-oriented extractive model that is based on distant supervision of a pointer–generator network leveraging bidirectional LSTMs and a copy mechanism and that is able to cope with the *out-of-vocabulary* phenomenon. The model has been trained on a large annotated corpus compiled specifically for this task from 250K Wikipedia pages, and tested on regular pages, where the pointers to other pages are considered as ground truth concepts. The outcome of the experiments shows that our model significantly outperforms standard techniques and, when used on top of DBpedia Spotlight, further improves its performance. The experiments furthermore show that the model can be readily ported to other datasets on which it equally achieves a state-of-the-art performance.

Keywords: Open-domain discourse texts · Concept extraction · Pointer-generator neural network · Distant supervision

1 Introduction

In knowledge discovery and representation, the notion of *concept* is most often used to refer to *sense*, i.e., 'abstract entity' or 'abstract object' in the Fregean dichotomy of *sense* vs. *reference* [10]. In Natural Language Processing (NLP), the task of detection of surface forms of concepts, namely *Concept Extraction* (CE), deals with the identification of the language side of the concept coin, i.e., Frege's *reference.* Halliday [16] offers a syntactic interpretation of *reference.* In his terminology, it is a "classifying nominal group". For instance, *renewable energy* or *nuclear energy* are classifying nominal groups: they denote a class (or type) of energy, while, e.g., *cheap energy* or *affordable energy* are not: they do not typify, but rather qualify *energy* (and are thus "qualifying nominal groups").

CE is crucial for a number of downstream applications, including, e.g., language understanding, ontology population, semantic search, and question answering; it is also the key to entity linking [22]. In generic open domain subject-neutral discourse across different (potentially unrelated) subjects, indexing the longest possible nominal chunks and their head words located in sequences of tokens between specified “break words” [34] and special dictionary lookups such as *DBpedia Spotlight* [6] and *WAT* [28] are very common techniques. They generally reach outstanding precision, but low recall due to constant evolvement of the language vocabulary. Advanced deep learning models that already dominate CE in specialized closed domain discourse on one or a limited range of related subjects, e.g., biomedical discourse [14, 33], and that are also standard in keyphrase extraction [2, 25] are an alternative. However, such models need a tremendous amount of labeled data for training.

We present an operational CE model that utilizes pointer–generator networks with bidirectional long short-term memory (LSTM) units [12, 30] to retrieve concepts from general discourse textual material.[3] Furthermore, since for a generic, open domain concept extraction model we need a sufficiently large training corpus that covers a vast variety of topics and no such annotated corpora are available, we opt for distant supervision to create a sufficiently large and diverse dataset. Distant supervision consists in automatic labeling of potentially useful data by an easy-to-handle (not necessarily accurate) algorithm to obtain an annotation which is likely to be noisy but, at the same time, to contain enough information to train a robust model [26]. Two labeling schemes are considered. Experiments carried out on a dataset of 250K+ Wikipedia pages show that copies of our model trained differently and joined in an ensemble significantly outperform standard techniques and, when used on top of DBpedia Spotlight, further improve its performance by nearly 10%.

2 Related work

In this section, we focus on the review of generic discourse CE; for a comprehensive review of the large body of work on specialized discourse CE, and, in particular, on biomedical CE; see, e.g., [15]. We also do not discuss recent advances in keyphrase extraction [2] because their applicability to generic concept extraction is limited due to the specificity of the task.

The traditional CE techniques interpret any single and multiple token nominal chunk as a concept [34] or do a dictionary lookup, as, e.g., *DBpedia Spotlight* [6], which matches and links identified nominal chunks with DBpedia entries (6.6M entities, 13 billion RDF triples)[4], based on the Apache OpenNLP[5] models for phrase chunking and named entity recognition (NER). Given the large coverage of DBpedia, the performance of DBpedia Spotlight is rather competitive. However, obviously, the presence of an entry cannot always be ensured. Consider, e.g., a paper title “Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing”, where DBpedia Spotlight does not detect “Bloom embeddings” or “incremental parsing”, as there are no such entries in DBpedia.

[3] We adopt Halliday’s notion of classifying nominal group as definition of a concept.

[4] https://wiki.dbpedia.org/develop/datasets/dbpedia-version-2016-10

[5] https://opennlp.apache.org/

As DBpedia Spotlight, AIDA [35] relies on an RDF repository, YAGO2. WAT and its predecessor TagMe [28] use a repository of possible spots made of wiki-anchors, titles, and redirect pages. Both TagMe and WAT rely on statistical attributes called *link probability* and *commonness*; WAT draws furthermore on a set of statistics to prune a set of mentions using an SVM classifier. Wikifier [4] focuses on relation extraction, relying on a NER, which uses gazetteers extracted from Wikipedia and simple regular expressions to combine several mentions into a single one. All of them are used for state-of-the-art entity linking and (potentially nested) entity mention detection and typing [17, 36]. FRED [11] also focuses on extraction of relations between entities, with frames [9] as the underlying theoretical constructs. Unlike Wikifier and FRED, e.g., OLLIE [24] does not rely on any precompiled repository. It outperforms its strong predecessors REVERB [8] in relation extraction by expanding the set of possible relations and including contextual information from the sentence from which the relations are extracted.

A number of works focus on the recognition of named entities, which are the most prominent concept type. NERs work at a sentence level and aim at labeling all occurred instances. Among them, Lample et al. [20] provide a state-of-the-art NER model that avoids traditional heavy use of hand-crafted features and domain-specific knowledge. The model is based on bidirectional LSTMs and Conditional Random Fields (CRFs) that rely on two sources of information on words: character-based word representations learned from an annotated corpus and unsupervised word representations learned from unannotated corpora. Another promising approach to NER is fine-tuning of a language representation model such as, e.g., BERT [7]. The pre-trained BERT model can be fine-tuned with just one additional output layer to create state-of-the-art models for a wide range of tasks, including NER, without substantial task-specific architecture modifications.

Pointer-generator networks, which are generally applied in summarization contexts, are also experimented with in information extraction; cf., e.g., [27], where they have been successfully applied to the detection of term definitions in sentences with a specific structure and their translation into Description Logics formulæ using syntactic transformation. As a matter of fact, this similar task partially motivated our choice to use pointer-generator networks for CE.

3 Description of the model

We implement a deep learning model and a large-scale annotation scheme for distant supervision to cope autonomously with dictionary-independent generic CE and to complement state-of-the-art lookup-based approaches in order to increase their recall. In addition, we would like our model to perform reasonably well on pure NER tasks with a small gap to models specifically tuned for the NER datasets. The model follows the well-established tendency in information extraction adopted for NER and extractive summarization and envisages CE as an attention-based sequence-to-sequence learning problem.

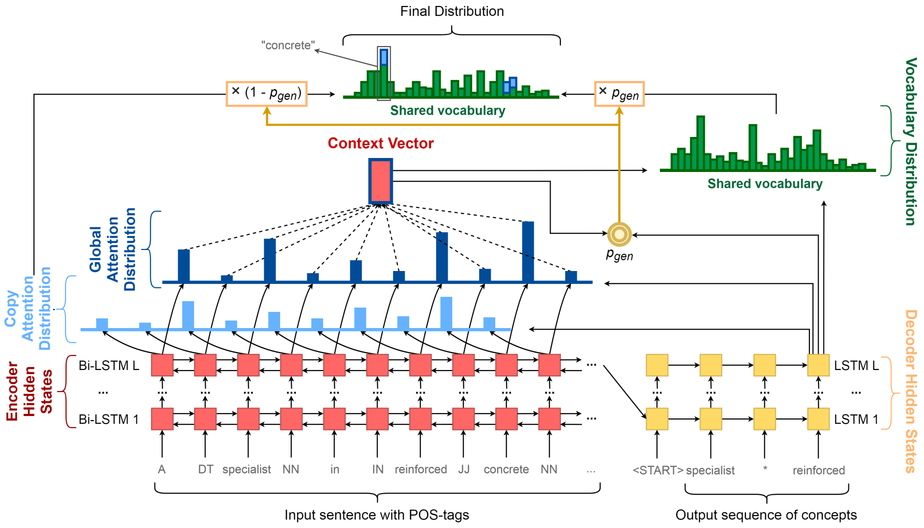

Fig. 1. The neural architecture for concept extraction

3.1 Overview of the model

As basis of our model, we use the pointer–generator network proposed in [30], which aids the creation of summaries with accurate reproduction of information. In each generation step t, the *pointer* allows for copying words w_i from the source sequence to the target sequence using the distribution of the attention layer a^t, while the *generator* samples tokens from the learned vocabulary distribution P_{vocab}, conditioned by a context vector h_t^* produced by the same attention layer, which is built based on hidden states h_i of an encoder (a bidirectional LSTM [12]) and states s_t of a decoder (a unidirectional LSTM). In addition, a coverage mechanism is applied to modify a^t using a coverage vector c^t to avoid undesirable repetitions in the output sequence. Specifically, to produce a word w, the above-mentioned distributions are combined into a single final probability distribution, which is weighted using the *generation probability* $p_{gen}\in$[0,1]:

$$P(w) = p_{gen} P_{vocab}(w) + (1 - p_{gen}) \sum\nolimits_{i:w_i=w} a_i^t \tag{1}$$

where $P_{vocab}(w)$ is the vocabulary distribution, which is zero if w is an out-of-vocabulary (OOV) word; a^t is the attention distribution; w_i - tokens of the input sequence; $\sum_{i:w_i=w} a_i^t$ is zero if w does not appear in the source sequence. In accordance with [30], the individual vectors, distributions, and probability p_{gen} are defined as follows:

$$c^t = \sum\nolimits_{t'=0}^{t-1} a^{t'} \tag{2}$$

$$e_i^t = v^T tanh(W_h h_i + W_s s_t + w_c c_i^t + b_{attn}) \tag{3}$$

$$a^t = softmax(e^t) \tag{4}$$

$$h_t^* = \sum\nolimits_i a_i^t h_i \tag{5}$$

$$P_{vocab} = softmax(V'(V[s_t, h_t^*] + b) + b') \tag{6}$$

$$p_{gen} = \sigma(w_{h*}^T h_t^* + w_s^T s_t + w_x^T x_t + b_{ptr}) \tag{7}$$

where v, W_h, W_s, w_c, b_{attn}, V, V', b, b', w_{h*}, w_s, w_x, b_{ptr} are learnable parameters, T stands for the transpose of a vector, x_t is the decoder input, and σ is the sigmoid function.

To adapt this basic model to the task of CE, we applied several modifications to it (cf., Figure 1[6]): (i) following Gu et al. [13], we use separate distributions for copy attention and global attention, instead of one for both; (ii) experiments have shown that encoders and decoders with several LSTM layers perform better than with a single layer, such that we work with multiple layer LSTMs; how many is determined using a development dataset; (iii) we adapt the forms of input and target sequences to the specifics of the task of CE. The input is comprised of tokens and their part-of-speech (PoS) tags (e.g., 'The DT President NN is VBZ elected VBD by IN a DT direct JJ vote NN'). The target sequence concatenates concepts in the order they appear in the text and separates them by a token "*" especially introduced to partition the output (e.g., 'President * direct vote').

This model is naturally applicable to the task of CE since it facilitates the selection and transfer of subsequences of tokens that form classifying nominal groups (= concepts) from a given source sequence of tokens (= text input) to the target sequence (= partitioned sequence of concepts). The pointer mechanism implies the ability to cope with OOV words, which is crucial for open domain CE, while the generator implies the ability to adjust vocabulary distribution for selecting the next word (which might be a termination token "*") based on a given context vector, which allows us to implicitly take into account the domain specifics and linguistic features that facilitate the task of CE. Furthermore, the vocabulary distribution update adds the possibility to vanish or strengthen the copy effect and thus learn to distinguish concepts with outer modifiers (such as, e.g.,"hot *air*", "[fully] crewed *aircraft*", "reinforced *group*") from multiword concepts (such as, e.g., "*hot air balloon*", "*unmanned aerial vehicle*", "*reinforced concrete*").

3.2 Training and applying the model

For training, token sequences are taken from annotated sentences (see the compilation of the annotated training dataset in Section 4.2 below) with a sliding overlapping window of a fixed maximum length (see the Experiments section), which is minimally expanded if needed in order not to deal with incomplete concepts at the borders. The trained model is applied to unseen sentences, which are also split into sequences of tokens with an overlapping window of the same size, without any expansion. Finally, the corresponding mentions in the plain

[6] We use a similar layout as in [30] for easier comparison of our extension with the original model.

text are determined since the output format does not include offsets. In particular, following [17], we find all possible matches for all detected concepts and then successively select non-nested concepts from the beginning to the end of the sentence, giving priority to the longest, in case of a multiple choice.

4 Datasets

In what follows, we describe the data and the procedure for their weak annotation to create extensive training and test datasets.

4.1 Data

We take a snapshot of the WordNet synset-typed[7] Wikipedia [29], from which we use the raw texts of the Wikipedia pages and text snippets of the links to other pages as ground truth concepts regardless their type; cf., Figure 2[8]. These links often share the headings of anchor pages, which are in most cases some real-world entities, cf., e.g., "Arthur Heurtley House", "Price Tower", etc. Sometimes, they are also lexical variations of terms behind the link, as, e.g., the highlighted link in the fragment "the two small *coastal battleships* General-Admiral Graf Apraxin and Admiral Senyavin" leads to the page named "Coastal defence ship".

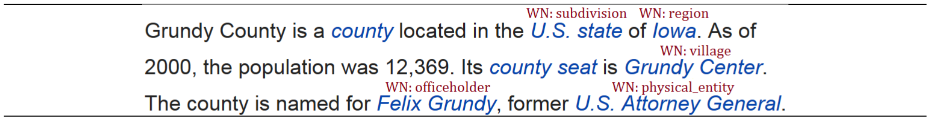
Grundy County is a *county* located in the *U.S. state* of *Iowa*. As of 2000, the population was 12,369. Its *county seat* is *Grundy Center*. The county is named for *Felix Grundy*, former *U.S. Attorney General*.
WN: subdivision
WN: region
WN: village
WN: officeholder
WN: physical_entity

Fig. 2. Ground truth concept annotation

The manual annotation of multi-word expressions in 100 randomly selected sentences with at least one multi-word link in each by a professional linguist showed that at least 63% of such phrases are indeed concepts (cf., e.g., "punctuated equilibrium", "chief of staff", "2004 presidential election"). For our work, we selected several data subsets from the collection of Wikipedia pages: 250 K pages to be weakly, but *densely* annotated.[9] Out of these 250K pages, 220K are used for training and 30K for validation. In addition, we use 7K Wikipedia pages with the sparse gold standard annotation as development set for choosing parameters of distant supervision and selecting the best model among several models

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

[8] Wikipedia does not contain self-links, therefore the concept *"Grundy County"* in a text from the self-titled page is not a link.

[9] Henceforth, we refer to the link snippet-based annotation of the pages as a *sparse* gold standard annotation since it covers by far not all concepts encountered in a page. Our distant supervision-based annotation is referred to as *dense* annotation since it (supposedly) covers all concepts on a given page. As usual, distant supervision-based annotation is also referred to as *weak* since it is an automatic annotation.

trained with different parameters, and 7K pages with the sparse gold standard annotation as test set. The test set does not allow an exhaustive evaluation of the model since it does not contain many generic concepts. However, given the lack of a large manually annotated dataset, this is still the best choice. Furthermore, in view of the fact that one third of the concepts in the test set are unseen during training, allows for the assessment of the ability of our model to handle OOV concepts.

4.2 Compilation of the training corpus

We automatically create a (noisy) training corpus using two various annotators over a large unlabeled dataset: DBpedia Spotlight with the value of its confidence coefficient, which gains the highest recall, and our own algorithm, which uses a number of rules and heuristics. Our labeling is based on the sentence-wise analysis of statistical and linguistic features of sequences of tokens. First, named entities and multiple token concepts and then single token concepts are identified. The algorithm covers the following tasks:

1. Application of a statistical NER model. A significant number of concepts in Wikipedia are capitalized terms, which can be captured by statistical named entity recognizers (NER); see Section 2 above. Therefore, at first, SpaCy's state-of-the-art NER model [18] is applied with a successive elimination of used tokens for further processing. The next steps are applied then separately to fragments of texts located between the identified NEs.

2. Selection of n-grams as fragments of noun phrase chunks that can form part of multiple token concepts. For this task, we define PoS-patterns based on Penn Treebank tagset[10], which were inherited from the patterns for multiword expression detection introduced in [5] and expanded here, resulting in the following set: P = {N_N, J_N, V_N, N_J, J_J, V_J, N_of_N, N_of_DT_N, N_of_J, N_of_DT_J, N_of_V, N_of_DT_V, CD_N, CD_J}, where N stands for "noun", i.e., NN|NNS|NNP|NNPS, J stands for "adjective", i.e., JJ| JJR|JJS, V – "verb" but limited to VBD|VBG|VN, CD – "cardinal number", DT – "determiner", and "of" is an exact pronoun. Each pattern matches an n-gram with two open-class lexical items and at most two auxiliary tokens between them.

3. Assessment of the distinctiveness of each selected n-gram. The distinctiveness of the selected n-grams is assessed using word co-occurrences from the Google Books N-gram Corpus [21]. Let us assume a given n-gram $T_1A_1A_2T_2 \in c_k$, where T_1 and T_2 are open class lexical items and A_1 and A_2 are optional auxiliary tokens, and c_k is a set of all n-grams of a particular kind of pattern $p_k \in P$. We use $T_1A_1A_2T_2$ as a point of a function that passes through normalized document frequencies of a set of similar n-grams $T_1A_1A_2T_j$, $j \in \{i \mid T_1A_1A_2T_i \in c_k\}$ arrayed in ascending order, to find a tangential angle at this point $\alpha_1 \in [0°; 90°)$. As an illustration, one may think of Zipf curves built from

[10] https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

the tail to the head individually for each set of n-grams. Similarly, $\alpha_2 \in [0°; 90°)$, is a tangential angle at the point $T_1 A_1 A_2 T_2$ on a curve of ordered frequencies of n-grams $T_h A_1 A_2 T_2$, $h \in \{i \mid T_i A_1 A_2 T_2 \in c_k\}$. We leverage these angles to check how prominent an n-gram is, i.e., to what extent it differs from its neighbors by overall usage. In case an n-gram is located among equally prominent n-grams with a tangential angle close to 0°, we do not consider it as a potential part of a concept since it does not show a notable distinctiveness inherent in concepts, especially in common idiosyncratic concepts. The thresholds α_{min_1} and α_{min_2} ($\alpha_{min_1} \geq \alpha_{min_2}$) for minimally allowed tangential angles such as $\max(\alpha_1, \alpha_2) \geq \alpha_{min_1}$, $\min(\alpha_1, \alpha_2) \geq \alpha_{min_2}$ are predermined in development experiments. We calculate tangential angles through central difference approximation with a coarse-grained grid:

$$\alpha = \arctan(\frac{f(x+h) - f(x-h)}{2h}) \cdot \frac{180}{\pi} \tag{8}$$

where h was chosen large enough ($h = 50$ in general, and it is maximum possible on the borders) for smoothing the curve to eliminate numerous abrupt changes in document frequency with relatively low amplitude. Thus, the approximation is intentionally carried out less accurately to result in such values that in practice form a curve with longer monotonous sections. Cf., Figure 3 for an example of assessing the prominence of an n-gram "prestressed concrete"; in the above notation, T_1 equals "prestressed_ADJ", A_1 and A_2 are omitted, and T_2 equals "concrete_NOUN".

Table 1. Tangential angles of concept candidates

Candidate	Angle	Wiki-term
reinforced_ADJ concrete_NOUN	89.77	YES
mixed_ADJ concrete_NOUN	89.07	NO
prestressed_ADJ concrete_NOUN	88.40	YES
pre-cast_ADJ concrete_NOUN	83.66	YES
first_ADJ concrete_NOUN	33.12	NO
original_ADJ concrete_NOUN	16.63	NO
massive_ADJ concrete_NOUN	9.85	NO
resistant_ADJ concrete_NOUN	8.08	NO
special_ADJ concrete_NOUN	5.66	NO
polymer_ADJ concrete_NOUN	4.03	YES
tall-wall_ADJ concrete_NOUN	1.90	NO
large_ADJ concrete_NOUN	1.75	NO
open_ADJ concrete_NOUN	0.75	NO
...	...	...
unusual_ADJ concrete_NOUN	OOV	NO
raised_ADJ concrete_NOUN	OOV	NO
...	...	...

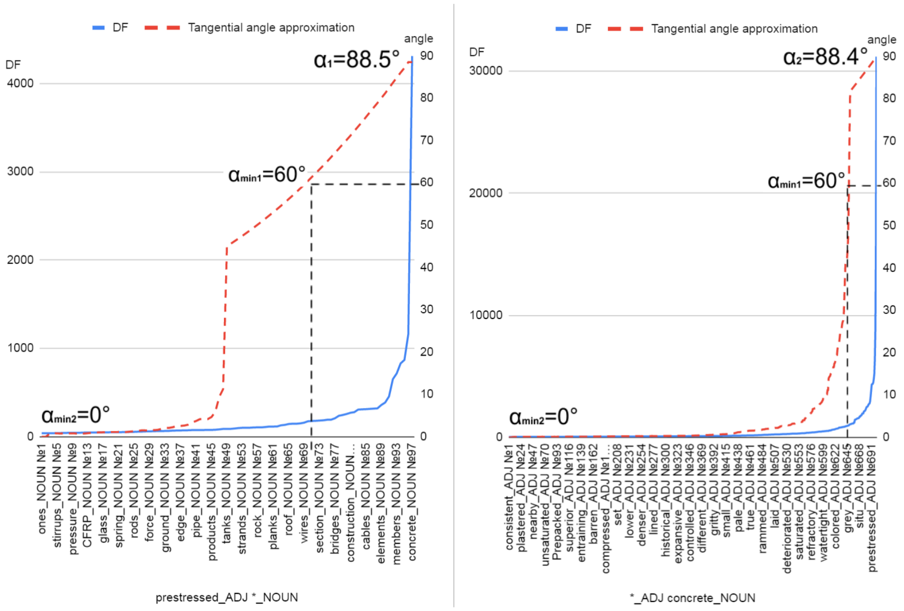

Fig. 3. Relation between document frequency and coarse-grained tangential angle approximation

Table 1 illustrates how the approximations of tangential angles differentiate classifying nominal groups from qualifying nominal groups. The most of the candidates with a large tangential angle have a separate article in Wikipedia (i.e., they are likely to be concepts), while candidates with a small tangential angle or without an entry in Google Books (and belong thus to OOV) in general do not have a Wikipedia article. This shows that the chosen criterion for differentiating the concepts is suitable for weak annotation within distant supervision.

Grid search was applied to find the best combination of parameters α_{min_1} and α_{min_2} from the three possible tangential angles: 85°, 60°, and 0°. These angles correspond to various levels of the distinctiveness of a concept and therefore give dissimilar annotations. As a result, $\alpha_{min_1} = 60°$ and $\alpha_{min_2} = 0°$ gave the best scores on the development set and were used for annotation of the training set.

4. Combination of intersected highly distinctive parts as concepts. We combine those distinctive n-grams that share common tokens and iteratively drop the last token in each group if it is not a noun, in order to end up with complete noun phrase candidate concepts (e.g., "value of the played card" is a potential concept corresponding to the patterns {N_of_DT_V; V_N}). Some single-word concepts already might appear at this point.

5. Recovery of missed single-word concepts. To enrich the set of candidate concepts, we consider all unused nouns and numbers in a text as single-word concept candidates.

The obtained training corpus contains moderate amount of noise: the proposed annotation algorithm outperforms some baselines and might be used for CE by itself; cf. setup (A) in Tables 2 and 3 with results of evaluation in the following section.

5 Experiments

5.1 Setup of the experiments

For our experiments, we use the realization of See et al. [30]'s pointer–generator model in the OpenNMT toolkit [19], which allows for the adaptation of the model to the task of CE along the lines described in Section 3.1 above. We use the default OpenNMT attention proposed in [23], which simplifies and generalizes the attention mechanism of [3] used in [30]. Furthermore, the default types of the alignment functions are used: *general* for copy attention and *dot* for global attention, as suggested in [23].

The model has 512-dimensional hidden states and 256-dimensional word embeddings shared between encoder and decoder. We use a vocabulary of 50k words as we rely mostly on a copying mechanism that uses dynamic vocabulary made up of words from the current source sequence. We train the CE-adapted pointer–generator networks of two and three bi-LSTM layers with 20K and 100K training steps on the two training datasets (obtained using Google Books and DBpedia Spotlight, respectively; see above) using the Stochastic Gradient Descent on a single GeForce GTX 1080 Ti GPU with a batch size of 64. Validation and saving of checkpoint models has been performed at each one-tenth of the number of training steps.

In order to compare our extended pointer–generator model with state-of-the-art techniques, several efficient entity extraction algorithms were chosen as baselines: OLLIE [24], AIDA [35], AutoPhrase+ [31], DBpedia Spotlight [6], WAT [28],[11] and several state-of-the-art NER models, namely SpaCy NER [18], FLAIR NER [1] and two deep learning-based NER models [7,20][12]. AutoPhrase+ was used in combination with the StanfordCoreNLP PoS-tagger (as it was reported to show better performance with PoS-tags) and trained separately on its default DBLP dataset and on the above-mentioned raw Wikipedia texts our training dataset is composed of. Its output was slightly modified by removing auxiliary tokens from the beginning and the end of the phrase to make it more competitive with the rest of the algorithms. OLLIE's and SpaCy's outcomes were also modified the same way, which improved their performance. DBpedia Spotlight was applied with two different values of confidence coefficient: 0.5 (default value) and 0.1, which increased the recall.

[11] FRED [11] was not used as baseline as it is not scalable enough for the task: its REST service has a strong limitation on a number of possible requests per day, and it fails on processing long sentences (approximately more than 40 tokens).

[12] https://github.com/glample/tagger; https://github.com/kyzhouhzau/BERT-NER

The performance is measured in terms of precision, recall, and F_1-score, aiming at high recall, first of all. Since positive ground truth examples are sparse, and there are no negative examples, we treated only the detected concepts that partially overlapped the ground truth concepts as false positives. Concepts that have the same spans as the ground truth concepts are counted as true positives, and missed ground truth concepts as false negatives. This perfectly meets our goal to detect the exact match. It also allows us to penalize brute force high-recall algorithms that produce a large number of nested concepts, which are of limited use in real-world applications. Table 2 shows the reached performance on the domain-specific datasets, and Table 3 on the open domain set. The sign "*" stands for modifications made on cutting some first and last words of detected concepts in order to present them as "canonic" noun phrases, and "**" stands for removing nested concepts when this procedure gave better scores.

Table 2. Results on the domain-specific datasets

		"Architecture"			"Terrorist groups"		
Setup	Model	P	R	F_1	P	R	F_1
	FLAIR (Akbik et al., 2019)	0.79	0.74	0.76	0.77	0.66	0.71
	BERT NER (Delvin et al., 2019)	0.78	0.74	0.76	0.78	0.67	0.72
	AutoPhrase+$^{**}_{DBLP}$ (Shang et al., 2018)	0.38	0.44	0.41	0.31	0.34	0.33
	AutoPhrase+$^{**}_{WIKI}$ (Shang et al., 2018)	0.42	0.52	0.46	0.36	0.45	0.40
	SpaCy NER (Honnibal and Montani, 2017)	0.59	0.51	0.55	0.5	0.41	0.45
	SpaCy NER* (Honnibal and Montani, 2017)	0.71	0.61	0.66	0.59	0.49	0.54
	NER Tagger (Lample et al., 2016)	0.78	0.71	0.75	0.76	0.65	0.7
	WAT** (Piccinno and Ferragina, 2014)	0.66	0.53	0.59	0.64	0.5	0.56
	Spotlight$_{0.5}$ (Daiber et al., 2013)	**0.85**	0.74	0.79	**0.8**	0.7	**0.75**
	Spotlight$_{0.1}$ (Daiber et al., 2013)	0.7	0.79	0.74	0.65	0.77	0.7
	OLLIE* (Schmitz et al., 2012)	0.46	0.2	0.28	0.41	0.22	0.28
	AIDA (Yosef et al., 2011)	0.76	0.57	0.65	0.74	0.54	0.62
(A)	DSA$_{(60,0)}$	0.63	0.74	0.68	0.5	0.64	0.56
(B)	$PG_{(3L,80K)}(DSA_{DICT})$	0.67	0.77	0.72	0.61	0.73	0.66
(C)	$PG_{(2L,18K)}(DSA_{(60,0)})$	0.7	0.8	0.75	0.59	0.72	0.65
(D)	(B) + (C)	0.75	0.83	0.79	0.66	0.77	0.71
(E)	(B) + (C) + Spotlight$_{0.1}$	0.78	0.85	0.81	0.7	**0.8**	**0.75**
(F)	(B) + (C) + Spotlight$_{0.5}$	0.78	0.85	0.81	0.7	**0.8**	**0.75**
(G)	(C) + Spotlight$_{0.5}$	0.79	**0.86**	**0.82**	0.69	0.79	0.74

Table 3 displays the scores of two different experiment runs. In the first, only concepts with an assigned WordNet type label in our typed Wikipedia dataset (in their majority, named entities; cf. [29] for details of the typification) were considered as positive examples (from about 276K nouns in the test set, only 83K nouns, i.e., about 30%, were part of ground truth concepts); in the second, all text snippets of the links were taken as ground truth concepts (from about 390K nouns in the test set, 141K nouns, i.e., about 36%, were part of ground truth concepts). Setups A – H display the performance of different variants of our model. When several outcomes are merged to check if one can benefit from a combination of various models (as in D-H), we follow "first the earliest, then the longest" strategy as in Section 3.2. 'DSA_{DICT}' stands for the distant supervision annotation obtained using DBpedia Spotlight, i.e., a dictionary lookup, while '$DSA_{(60,0)}$' – for the proposed token-cooccurrence frequency-based method (cf. Step 3 of the compilation of the training corpus), where the values in parentheses correspond to α_{min_1} and α_{min_2}, which gave the best scores on the development set. $PG_{(2L,18K)}$ and $PG_{(3L,80K)}$ stand for pointer–generator networks with parameters shown in parentheses chosen using the development set (2 layers, $18K/20K$ training steps and 3 layers, $80K/100K$ training steps).

Table 3. Results on a large-scale open-domain dataset

		Only WordNet-typed concepts			All ground truth concepts		
Setup	Model	P	R	F_1	P	R	F_1
	FLAIR (Akbik et al., 2019)	**0.8**	0.74	0.77	**0.79**	0.59	0.67
	AutoPhrase+$^{**}_{DBLP}$ (Shang et al., 2018)	0.42	0.45	0.43	0.4	0.43	0.41
	AutoPhrase+$^{**}_{WIKI}$ (Shang et al., 2018)	0.46	0.52	0.49	0.43	0.49	0.46
	NER Tagger (Lample et al., 2016)	0.78	0.72	0.75	0.77	0.58	0.66
	WAT** (Piccinno and Ferragina, 2014)	0.72	0.55	0.62	0.68	0.42	0.52
	Spotlight$_{0.1}$ (Daiber et al., 2013)	0.73	0.76	0.75	0.69	0.73	0.71
	OLLIE* (Schmitz et al., 2012)	0.45	0.19	0.27	0.44	0.18	0.26
	AIDA (Yosef et al., 2011)	**0.8**	0.6	0.68	0.77	0.45	0.57
(A)	DSA$_{(60,0)}$	0.68	0.75	0.71	0.65	0.72	0.68
(B)	$PG_{(3L,80K)}(DSA_{DICT})$	0.71	0.74	0.73	0.68	0.72	0.7
(C)	$PG_{(2L,18K)}(DSA_{(60,0)})$	0.71	0.81	0.76	0.68	0.76	0.72
(D)	(B) + (C)	0.76	0.84	0.8	0.72	0.8	0.76
(H)	(C) + Spotlight$_{0.1}$	0.78	**0.85**	**0.81**	0.75	**0.81**	**0.78**

To compare the performance of our model with state-of-the-art NER, we applied it to two common public datasets for NER (CoNLL-2003 and GENIA). Table 4 shows the results on the CoNLL-2003 dataset for two variants of our model (Setups B and C) trained on our large training set, without any further NER adaptation, as well as for their updated versions (Setups I, J, and K), which were fine-tuned with the training set of the shared task $CoNLL_T$, contrasted with the results of the two genuine state-of-the-art NE recognizers [20] and [7] and DBpedia Spotlight. It should be noted that NER is a concept extraction subtask which aims at detecting less generic concepts. Consider the following statistics that highlight the difference of NER with generic CE: from about 69K nouns in the CoNLL-2003 training set, only 31K nouns are part of NEs (e.g., S&P, BAYERISCHE VEREINSBANK, London Newsroom, Lloyds Shipping Intelligence Service), while the remaining 38K nouns (as in "air force", "deposit rates", "blue collar workers") are not part of NEs; as far as GENIA is concerned, from about 132K nouns, only 93K form NEs (e.g., "tumor necrosis factor", "terminal differentiation", "isolated polyclonal B lymphocytes"), while the remaining 39K do not (as "colonies", "interpretation", "notion", "circular dichroism", "differential accumulation").

Table 4. Results on the CoNLL-2003 datasets

		CoNLL-2003 (test-a)			CoNLL-2003 (test-b)		
Setup	Model	P	R	F_1	P	R	F_1
	BERT NER (Delvin et al., 2019)	0.95	0.96	0.95	0.94	0.94	0.94
	NER Tagger (Lample et al., 2016)	0.97	0.97	**0.97**	0.97	0.96	**0.96**
	Spotlight$_{0.5}$ (Daiber et al., 2011)	0.9	0.63	0.74	0.9	0.65	0.75
	Spotlight$_{0.1}$ (Daiber et al., 2011)	0.77	0.77	0.77	0.76	0.77	0.77
(B)	$PG_{(3L,80K)}(DSA_{DICT})$	0.81	0.78	0.8	0.81	0.79	0.8
(C)	$PG_{(2L,18K)}(DSA_{(60,0)})$	0.82	0.82	0.82	0.79	0.81	0.8
(I)	$FineTune((B), CoNLL_T)$	0.95	0.92	0.93	0.95	0.92	0.94
(J)	$FineTune((C), CoNLL_T)$	0.94	0.91	0.93	0.96	0.92	0.94
(K)	(I) + (J)	0.94	0.93	0.93	0.95	0.93	0.94

Table 5 shows the results of our models fine-tuned with GENIA along with the results of concept identification by the recently published model [32],[13] which provides the most promising scores on different GENIA tasks.

Table 5. Results on the GENIA dataset

		GENIA		
Setup	Model	P	R	F_1
	seq2seq (Straková et al., 2019)	0.86	0.79	0.82
(L)	$FineTune((B), GENIA_T)$	0.85	0.8	0.82
(M)	$FineTune((C), GENIA_T)$	0.84	0.77	0.81
(N)	(L) + (M)	0.85	0.8	**0.83**

5.2 Discussion

Tables 2 and 3 show that a combination of the different variants of the proposed pointer-generator model, which do not rely on external dictionaries after being trained (cf. Setup D), outperforms in terms of recall and F_1-score nearly all other models, including the dictionary lookup-based DBpedia Spotlight, which is a hard to beat as it was applied to "known" data. However, a combination of the pointer–generator model with DBpedia Spotlight is even better; it outperforms DBpedia Spotlight by 10%. In other words, a deep model combined with a DBpedia-lookup is the best solution for generic CE. This applies to both runs displayed in Table 3, while all tested models show a lower performance in the

[13] https://github.com/ufal/acl2019_nested_ner

discovery of non-named entities. In particular, the NER models expectedly suffer a dramatic drop in recall. In terms of precision, DBpedia Spotlight on its own is considerably better than any other proposal on the two small domain-specific test sets, while AIDA is best on the open domain test set. This is to be expected for dictionary lookup-based strategies. Also, as expected, DBpedia Spotlight, applied with its confidence coefficient = 0.1, showed significantly better recall than with the default value of 0.5, although the F_1-score was lower. The experiment on the CoNLL-2003 dataset shows that our model for generic CE performs well even without any special adjustment ($F_1 = 0.8 - 0.82$). It can be further fine-tuned to the specific dataset resulting in scores comparable to state of the art, even if not designed specifically for the NER task ($F_1 = 0.93 - 0.94$), while its overall CE performance is better than of the targeted NER models (compare, e.g., (B)+(C) with Lample et al. (2016)'s NER in Tables 2 and 3.

We also assessed the ability of our model to detect OOV concepts taking Setup C as an example. We found out that it detected correctly 87% of known concepts and roughly 50% of the concepts unseen during the training phase. The latter include such entities as "bertsolaritza", "rotary table", "oil refining complex", "rope ferry", "Lake of Two Mountains", "Gyrodyne Company of America", etc. Concepts that were missed often have unusual structures in terms of PoS-tag sequences or ways of capitalization; cf., e.g., "As the Rush Comes" (detected as "Rush Comes"), and "New York Times Co. v. Sullivan" (detected as "New York Times Co.", "v.", "Sullivan").

6 Conclusions

We presented an adaptation of the pointer–generator network model [30] to generic open domain concept extraction. Due to its capacity to cope with OOV concept labels, the model outperforms dictionary lookup-based CE such as DBpedia Spotlight or AIDA in terms of recall and F_1-score. It also shows an advantage over deep models that focus on NER only since it also covers non-named concept categories. However, a combination of the pointer–generator model with DBpedia Spotlight seems to be the best solution since it takes advantage of both the neural model and the dictionary lookup. In order to facilitate a solid evaluation of the proposed model and compare it to a series of baselines, we utilized Wikipedia pages with text snippet links as a sparsely concept-annotated dataset. To ensure that our model is capable of extracting all generic concepts instead of detecting only texts of the page links, we ignored this sparse annotation during training. Instead, we compiled a large densely concept-annotated dataset for leveraging it within the distant supervision using the algorithm described above. To the best of our knowledge, no such dataset was available so far. In the future, we plan to address the problem of multilingual concept extraction, using pre-trained multi-lingual embeddings and compiling another large dataset that contains a higher percentage of non-named entity concepts.

The code for running our pretrained models is available in the following GitHub repository: https://github.com/TalnUPF/ConceptExtraction/.

7 Acknowledgments

The work presented in this paper has been supported by the European Commission within its H2020 Research Programme under the grant numbers 700024, 700475, 779962, 786731, 825079, and 870930.

References

1. Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., Vollgraf, R.: FLAIR: An easy-to-use framework for state-of-the-art NLP. In: Proc. NAACL (2019)
2. Al-Zaidy, R., Caragea, C., Giles, C.L.: Bi-lstm-crf sequence labeling for keyphrase extraction from scholarly documents. In: Proc. of WWW (2019)
3. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: 3rd International Conference on Learning Representations (2015)
4. Cheng, X., Roth, D.: Relational inference for wikification. In: Proc. of the EMNLP. pp. 1787–1796 (2013)
5. Cordeiro, S., Ramisch, C., Villavicencio, A.: Ufrgs&lif at semeval-2016 task 10: rule-based mwe identification and predominant-supersense tagging. In: Proc. of SemEval-2016. pp. 910–917 (2016)
6. Daiber, J., Jakob, M., Hokamp, C., Mendes, P.: Improving efficiency and accuracy in multilingual entity extraction. In: Proc. of the 9th International Conference on Semantic Systems (I-Semantics) (2013)
7. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proc. of the NAACL-HLT. pp. 4171–4186 (2019)
8. Fader, A., Soderland, S., Etzioni, O.: Identifying relations for open information extraction. In: Proc. of the EMNLP. pp. 1535–1545 (2011)
9. Fillmore, C., Baker, C.: Frame semantics for text understanding. In: Proc. of the JWordNet and Other Lexical Resources Workshop at NAACL (2001)
10. Frege, G.: Ueber Sinn und Bedeutung. Zeitschrift fuer Philosophie und philosophische Kritik **100**, 25–50 (1892)
11. Gangemi, A., Presutti, V., Reforgiato Recupero, D., Nuzzolese, A., Draicchio, F., Mongiovì, M.: Semantic web machine reading with fred. Semantic Web **8**(6), 873–893 (2017)
12. Graves, A., Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures. Neural networks **18**(5-6), 602–610 (2005)
13. Gu, J., Lu, Z., Li, H., Li, V.O.: Incorporating copying mechanism in sequence-to-sequence learning. In: Proc. of the ACL. pp. 1631–1640 (2016)
14. Habibi, M., Weber, L., Neves, M., Wiegandt, D.L., Leser, U.: Deep learning with word embeddings improves biomedical named entity recognition. Bioinformatics **33**(14), i37–i48 (2017)
15. Hailu, N.G.: Investigation of Traditional and Deep Neural Sequence Models for Biomedical Concept Recognition. Ph.D. thesis, University of Colorado (2019)
16. Halliday, M.: Halliday's Introduction to Functional Grammar. Routledge, London & New York (2013)
17. Hasibi, F., Balog, K., Bratsberg, S.: Entity linking in queries: Tasks and evaluation. In: Proc. International Conference on The Theory of Information Retrieval. pp. 171–180. ACM (2015)

18. Honnibal, M., Montani, I.: spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. https://spacy.io/ (2017)
19. Klein, G., Kim, Y., Deng, Y., Nguyen, V., Senellart, J., Rush, A.: Opennmt: Neural machine translation toolkit. In: Proc. of the 13th Conference of the AMTA. vol. 1, pp. 177–184 (2018)
20. Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., Dyer, C.: Neural architectures for named entity recognition. In: Proc. NAACL-HLT (2016)
21. Lin, Y., Michel, J.B., Aiden Lieberman, E., Orwant, J., Brockman, W., Petrov, S.: Syntactic annotations for the Google books NGram corpus. In: Proc. of the ACL 2012 System Demonstrations. pp. 169–174 (Jul 2012)
22. Logeswaran, L., Chang, M.W., Lee, K., Toutanova, K., Devlin, J., Lee, H.: Zero-shot entity linking by reading entity descriptions. In: Proc. of the ACL. pp. 3449–3460 (Jul 2019)
23. Luong, T., Pham, H., Manning, C.: Effective approaches to attention-based neural machine translation. In: Proc. of the EMNLP. pp. 1412–1421 (2015)
24. Mausam, Schmitz, M., Soderland, S., Bart, R., Etzioni, O.: Open language learning for information extraction. In: Proc. of the 2012 Joint EMNLP and CoNLL Conferences. pp. 523–534 (2012)
25. Meng, R., Zhao, S., Han, S., He, D., Brusilovsky, P., Chi, Y.: Deep keyphrase generation. In: Proc. ACL. pp. 582–592 (2017)
26. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proc. of the ACL. pp. 1003–1011 (2009)
27. Petrucci, G., Rospocher, M., Ghidini, C.: Expressive ontology learning as neural machine translation. Journal of Web Semantics **52**, 66–82 (2018)
28. Piccinno, F., Ferragina, P.: From tagme to wat: A new entity annotator. In: Proc. of the First International Workshop on Entity Recognition and Disambiguation. pp. 55–62. ERD '14, ACM, New York, NY, USA (2014)
29. Schenkel, R., Suchanek, F., Kasneci, G.: Yawn: A semantically annotated wikipedia xml corpus. Datenbanksysteme in Business, Technologie und Web, –12. (2007)
30. See, A., Liu, P.J., Manning, C.D.: Get to the point: Summarization with pointer-generator networks. In: Proc. of the ACL. pp. 1073–1083 (2017)
31. Shang, J., Liu, J., Jiang, M., Ren, X., Voss, C., Han, J.: Automated phrase mining from massive text corpora. IEEE Transactions on Knowledge and Data Engineering **30**(10), 1825–1837 (2018)
32. Straková, J., Straka, M., Hajic, J.: Neural architectures for nested ner through linearization. In: Proceedings of the ACL. pp. 5326–5331 (2019)
33. Tulkens, S., Šuster, S., Daelemans, W.: Unsupervised concept extraction from clinical text through semantic composition. Journal of Biomedical informatics **91**, 103–120 (2019)
34. Woods, W.A.: Conceptual indexing: A better way to organize knowledge. Technical Report SMLI, TR97-61, Sun Microsystems Laboratories (1997)
35. Yosef, M., Hoffart, J., Bordino, I., Spaniol, M., Weikum, G.: Aida: An online tool for accurate disambiguation of named entities in text and tables. Proc. of the VLDB Endowment **4**(12), 1450–1453 (2011)
36. Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., Liu, Q.: ERNIE: Enhanced language representation with informative entities. In: Proc. ACL. pp. 1441–1451 (2019)