A Comparison of Context-sensitive Models for Lexical Substitution

Aina Garí Soler[1], Anne Cocos[2], Marianna Apidianaki[1,3], Chris Callison-Burch[2]

[1]LIMSI, CNRS, Univ. Paris Sud, Univ. Paris Saclay
[2]Department of Computer and Information Science, University of Pennsylvania
[3]LLF, CNRS, Univ. Paris Diderot
aina.gari@limsi.fr, acocos@seas.upenn.edu,
marianna@limsi.fr, ccb@seas.upenn.edu

Abstract

Word embedding representations provide good estimates of word meaning and give state-of-the art performance in semantic tasks. Embedding approaches differ as to whether and how they account for the context surrounding a word. We present a comparison of different word and context representations on the task of proposing substitutes for a target word in context (lexical substitution). We also experiment with tuning contextualized word embeddings on a dataset of sense-specific instances for each target word. We show that powerful contextualized word representations, which give high performance in several semantics-related tasks, deal less well with the subtle in-context similarity relationships needed for substitution. This is better handled by models trained with this objective in mind, where the inter-dependence between word and context representations is explicitly modeled during training.

1 Introduction

Contextualized word representations model complex characteristics of word usage, and give state-of-the-art performance in a variety of NLP tasks involving syntactic and semantic processing. Each proposed model accounts for context in a different way depending on the underlying architecture, and might account for local or long-distance phenomena. In this work, we compare different word representations on the lexical substitution (LexSub) task, which involves proposing meaning-preserving substitutes for words in specific contexts (McCarthy and Navigli, 2007). The importance of context in defining the meaning of word instances, and selecting the substitutes that best fit specific sentences, makes of the LexSub task an ideal testbed for a direct comparison of the contextualized representations built by different models.

We compare representations that model context in different ways: they exploit context embeddings generated within the skip-gram model (Melamud et al., 2015), learn a generic context embedding function using a bidirectional Long Short-Term Memory (LSTM) network (Melamud et al., 2016), or use vectors that are learned functions of the internal states of a deep bidirectional language model (biLM) (Peters et al., 2018a). Additionally, we experiment with a way to tune these state-of-the-art context-sensitive representations to sense-specific contexts of use, using a dataset of sentences containing each LexSub target word that are carefully chosen to reflect the senses of their potential substitutes. We explore the impact of this tuning on the LexSub task. Finally, we compare the performance of contextualized models to baseline models that exploit standard word embedding representations for measuring semantic similarity without directly accounting for context, such as Glove (Pennington et al., 2014) and FastText (Mikolov et al., 2018).

The results of this study highlight the importance of the architecture used for model training in capturing information relevant for lexical substitution. We show that contextualized representations that

Substitutes	Sentences
shoot (5)	The panther **fired** at the bridge and hit a truck.
sack (5), dismiss (1)	While both he and the White House deny he was **fired**, Frum is so insistent on the fact that he quit on his own that it really makes you wonder.
trainer (3), teacher (2), instructor (1), tutor (1)	As a **coach**, we speak and listen with the intent of helping people surface, question and reframe assumptions.
bus (5), carriage (1)	We hopped back onto the **coach** - now for the boulangerie!

Table 1: Examples of manually proposed substitutes for the verb *fire* and the noun *coach* in the SemEval-2007 Lexical Substitution dataset (McCarthy and Navigli, 2007). Numbers in brackets indicate the number of annotators who proposed each substitute.

have been shown to be very powerful in other semantics-related tasks perform less well in the LexSub task, while others that explicitly model the inter-dependence of words and their context manage to propose the best substitutes as measured by comparing their choices to human annotations in a gold standard dataset.

2 Related Work

The lexical substitution task consists in selecting meaning-preserving substitutes for words in context. Initially proposed as a testbed for word sense disambiguation systems (McCarthy and Navigli, 2007), in recent works it is mainly seen as a way of evaluating the in-context lexical inference capacity of vector-space models without explicitly accounting for sense (Kremer et al., 2014; Melamud et al., 2015). Examples of substitutes of words in context proposed by annotators in the SemEval-2007 Lexical Substitution dataset are presented in Table 1. The main idea behind these sense-unaware models is that the basic (out-of-context) representation of a word is adapted to each specific context of use. This is done by combining the basic vector of the word with the vectors of words found in its immediate context, or having a specific syntactic relation. Appropriate substitutes are synonyms or paraphrases of the word that are similar to this contextualized representation.

Melamud et al. (2015) use word embeddings generated using the word2vec skip-gram model (Mikolov et al., 2013). word2vec learns for every word type two distinct representations, one as a target and another as a context, both embedded in the same space. The context representations are generally discarded after training, considered internal to the model, and the output word embeddings represent context-insensitive target word types. Melamud et al. use the context embeddings in conjunction with the target word embeddings to model word instances in context, identify appropriate substitutes by measuring their similarity to the target and the context, and obtain state-of-the-art results on the LexSub task.

In later work, Melamud et al. (2016) propose *context2vec*, a model that uses a neural network architecture based on word2vec CBOW (Mikolov et al., 2013). context2vec replaces CBOW's representation of a word's surrounding context as a simple average of the embeddings of the context words in a fixed window, with a full sentence neural representation of context obtained using a bidirectional LSTM. Sentential contexts and target words are embedded in the same low-dimensional space, which is optimized to reflect inter-dependencies between them. This rich representation gives context2vec high performance in tasks involving context, such as lexical substitution, word sense disambiguation and sentence completion.

Peters et al. (2018a) propose a new type of deep contextualized word representations called *ELMo* (Embeddings from Language Models), where each token is assigned a representation that is a function of the entire input sentence. Vectors are derived from a bidirectional LSTM that is trained with a coupled language model (LM) objective on a large test corpus. ELMo representations are deep in the sense that they are a function of all of the internal layers of the biLM, which improves performance in several syntax and semantics-related tasks compared to using the top LSTM layer. The best combination of

layers is learnt jointly with a supervised NLP task. An analysis on different tasks shows that lower layers efficiently encode syntactic information, while higher layers capture semantics (Peters et al., 2018b). The gains observed in syntactic tasks outweigh those on semantic-related tasks, such as coreference resolution, Semantic Role Labeling and word sense disambiguation. In this work, we apply the ELMo vectors for the first time to the lexical substitution task and compare their performance to the context-sensitive models of Melamud et al. (2015) and Melamud et al. (2016). We also propose a way to tune the ELMo representations to the LexSub task, by using a dataset containing a high number of sentences for words in context that represent meanings close to that of their possible substitutes.

3 Substitute-focused Contexts

Contextualized word embeddings for a given target word vary based on the sense of a target word instance. Unlike the variation in discrete sense-level embeddings (e.g. Iacobacci et al. (2015); Rothe and Schütze (2015); Flekova and Gurevych (2016), and others), this variation is continuous. One of our experiments aims to see whether incorporating discrete fine-grained sense information into our LexSub models, where senses are defined at the level of substitute paraphrases, can improve performance. For this purpose, we generate a dataset of "focused contexts" (hereafter abbreviated FC) for each target word which are specifically chosen to represent the specific sense that target word shares with each of its potential substitutes.

The starting point for our focused contexts dataset is the Paraphrase Database (PPDB) (Ganitkevitch et al., 2013; Pavlick et al., 2015), a collection of over 80M English paraphrase pairs. PPDB was automatically built using the pivot method (Bannard and Callison-Burch, 2005), which discovers same-language paraphrases by 'pivoting' over bilingual parallel corpora. Specifically, if two English phrases such as "*under control*" and "*in check*" are each translated to the same German phrase "*unter kontrolle*" in some contexts, then this is taken as evidence that "*under control*" and "*in check*" have approximately similar meaning. Because PPDB was constructed using the pivot method, it follows that each paraphrase pair $x \leftrightarrow y$ in PPDB has a set of shared foreign translations. This idea is core to the method for extracting substitute-focused sentences.

The sentences for paraphrase pair $x \leftrightarrow y$ are extracted from the English side of English-to-foreign bitext corpora as follows. We assume there exists some set F^{xy} of foreign phrases to which x and y have both been independently translated. To find sentences containing x that correspond to its sense as a paraphrase of y, we simply enumerate English sentences containing x from the parallel corpora where x is aligned to some $f \in F^{xy}$. Sentences for y are extracted symmetrically. We refer to the set of English sentences containing x as $S^{\dot{x}y}$, and the set of English sentences containing y as $S^{x\dot{y}}$. Note that for some other paraphrase pair involving x, say $x \leftrightarrow z$, there may be sentences that appear in both $S^{\dot{x}y}$ and $S^{\dot{x}z}$ if their sets of shared translations, F^{xy} and F^{xz}, overlap.

Intuitively, we would like the sentences containing x in $S^{\dot{x}y}$ to be "highly characteristic" of the meaning of y, and vice versa. However, not all pivot translations $f \in F^{xy}$ produce equally characteristic sentences. For example, consider the paraphrase pair *bug* $\leftrightarrow$ *worm*. Their shared translation set, $F^{bug,worm}$, includes the French terms *ver* (*worm*) and *espèce* (*species*), and the Chinese term 虫 (*bug*). In selecting sentences for $S^{\dot{bug},worm}$, the FC dataset should prioritize English sentences where *bug* has been translated to the most characteristic translation for *worm* – *ver* – over the more general 虫 or *espèce*.

The degree to which a foreign translation is "characteristic" of an English term can be quantified by the pointwise mutual information (PMI) of the English term with the foreign term. To avoid unwanted biases that might arise from the uneven distribution of languages present in our bitext corpora, we treat PMI as language-specific. Given language l containing foreign words $f \in l$, we use shorthand notation f_l to indicate that f comes from language l. The PMI of English term e with foreign word f_l can be computed as:

$$\text{PMI}(e, f_l) = \frac{p(e, f_l)}{p(e) \cdot p(f_l)} = \frac{p(f_l|e)}{p(f_l)}$$

Substitutes	Substitute-focused sentences
sack	Yet what are proclamations on employment rights worth, when company bosses have a 'divine right' to hire and **fire**?
dismiss	They chose to **fire** a lot of people; to throw people out who weren't needed.
shoot	We hope that the generals and civilian oligarchs will not **fire** on the honduran people.
launch	A security source said electrical wiring found at the site suggested plans to **fire** the rockets by remote control.

Table 2: Examples of substitute-focused sentences for the verb *fire* corresponding to its substitutes.

The term in the numerator is the translation probability $p(f_l|e)$, which indicates the likelihood that English word e is translated to foreign term f_l in an English-l parallel corpus. Maximizing this term promotes the most frequent foreign translations for e. It is calculated as:

$$p(f_l|e) = \frac{\text{count}(e \rightarrow f_l)}{\sum_{f' \in l} \text{count}(e \rightarrow f')}$$

where $(e \rightarrow f_l)$ indicates the event that e is aligned to f_l in a bitext sentence pair. The term in the denominator is the likelihood of the foreign word, $p(f_l)$. Dividing by this term down-weights the emphasis on frequent foreign words. This is especially helpful for mitigating errors due to mis-alignments of English words with foreign stop words. The foreign word probability is calculated as:

$$p(f_l) = \frac{\text{count}(f_l)}{\sum_{f' \in l} \text{count}(f')}$$

To extract $S^{\dot{x}y}$, the set of English sentences containing x for paraphrase pair $x \leftrightarrow y$, we first order their shared translations, $f \in F^{xy}$, by decreasing $PMI(y, f)$. Then, for each translation f in order, we extract up to 2500 sentences from the bitext corpora where x is translated to f. This process continues until a maximum of 10k sentences containing x are generated. As a result of selecting sentences containing x in decreasing order of $PMI(y, f)$, the dataset includes contexts where the sense of x is most closely related to its paraphrase y.

To compile our dataset, we select sentences pertaining to all paraphrases of each target word in the LexSub datset. We extract sentences from the same English-to-foreign bitext corpora used to generate English PPDB (Ganitkevitch et al., 2013).

3.1 Deriving contextualized vectors from focused contexts

The focused context dataset groups sentences where a target word appears with a specific meaning, that of one of its paraphrases (possible substitutes) in PPDB. This makes the resource useful for lexical substitution, as it provides numerous examples of sentences for each target-substitute pair. In Table 2, we give examples of sentences for the word *fire* and its candidate substitutes (*sack*, *dismiss*, *shoot*, *launch*).

We use the sets of sentences available for each target-substitute pair to create contextualized representations for the candidate substitutes, using the approach proposed by Peters et al. (2018a) for applying the biLM representations to a supervised word sense disambiguation task. More precisely, we tune pretrained contextualized (ELMo) embeddings to the LexSub task using contexts from the FC dataset. A representation for a substitute of a target word is the average of the ELMo vectors obtained from the FC sentences corresponding to that substitute. For each substitute, we use the 100 sentences with the highest PMI, avoiding sentences with a high overlap in words.[1] The ELMo language model contains three layers, so each token in text has three different representations, one per layer. It is important to note that we do not train a neural model on this dataset, so we do not learn a linear combination of the biLM layers in the way ELMo is typically used. Instead, we experiment with the top layer (*FC-ELMo-top*) and

[1] We use an overlap threshold of 60%. This cleaning serves to discard highly similar sentences and ensure a varied vocabulary in the retained dataset. If for some substitutes less than 100 sentences are available after this filtering, we keep them all.

an average of the three layers (*FC-ELMo-avg*) of the biLM (5.5B) released by Peters et al. (2018a)[2]. We also use FC to tune context2vec embeddings released by Melamud et al. (2016) and pre-trained on the UkWac corpus[3] (*FC-c2v*). We create context representations from the high quality sentences retained for a target-substitute pair by replacing the target word with a blank slot. A representation for the substitute is then created by taking the average of all generated context representations. The obtained candidate vectors are used in the lexical substitution methods described in Section 4.

4 Lexical Substitution Methods

We present a head-to-head comparison of different context representations on the LexSub task. We evaluate all models on the SemEval Lexical Substitution task test set (McCarthy and Navigli, 2007). Given an instance of a target word t and a set of candidate substitutes ($S = \{s_1, s_2, ..., s_n\}$), each model provides a ranking of the substitutes depending on how well they describe the meaning of t in each specific sentence. Higher ranked substitutes are both good paraphrases of the target and a good fit in the context. In our experiments, candidate substitutes $S = \{s_1, s_2, ..., s_n\}$ for a target word t are its paraphrases in the Paraphrase Database (PPDB) XXL package (Pavlick et al., 2015)[4] that are also present in the gold standard annotations. This is a ranking variant of the LexSub task where systems are not expected to identify substitutes from the whole vocabulary, but rather to estimate the suitability of items in a specific pool of substitutes and rank them accordingly (Kremer et al., 2014). In what follows, we describe how the different methods represent words and contexts, and perform substitute ranking for new instances. An illustration of the different methods can be found in Figure 1.

4.1 Target-to-substitute similarity

ELMo representations are contextualized, in the sense that the embedding of a token is a function of the full sentence in which it appears. We propose a substitute ranking method that uses target-to-substitute (*tTs*) similarity, as measured by the cosine similarity of the corresponding ELMo representations. We use the top layer (*ELMo-top*) and the average of the three layers (*ELMo-avg*) of the biLM (5.5B) (Peters et al., 2018a) in the following way.

Given a new sentence C with an instance of the target word to be substituted, we first obtain an ELMo representation from this context corresponding to the target word. Then, we replace the target with all its potential substitutes, one at a time, and obtain the ELMo vector for each substitute in the context of C by feeding the new sentence as input to the biLM. Substitutes are then ranked by the cosine similarity of the target word's ELMo vector in C with that of the ELMo vector of each substitute in the same context.

We use this method with FC-ELMo as well. For each sentence, possible substitutes are ranked according to the similarity of their FC-ELMo embedding to the ELMo embedding of the target word in the sentence. We expect context to be indirectly taken into account by using such contextualized representations.

4.2 *AddCos*: *skip-gram* word and context embeddings

Melamud et al. (2015)'s method for lexical substitution is based on the skip-gram word embedding model. The novelty of the approach is that it explicitly leverages the context embeddings generated within skip-gram, generally considered as internal and discarded at the end of the learning process. The proposed context-sensitive substitutability measures for potential substitutes reflect a combination of two types of similarity: a) *target-to-substitute*, showing how similar a potential substitute is to the target word, and b) *target-to-context*, reflecting the substitute's compatibility with a given sentential context. Similarities are estimated using the vector Cosine distance between the respective skip-gram word and

[2]https://allennlp.org/elmo

[3]http://u.cs.biu.ac.il/ nlp/resources/downloads/context2vec/

[4]`http://paraphrase.org`

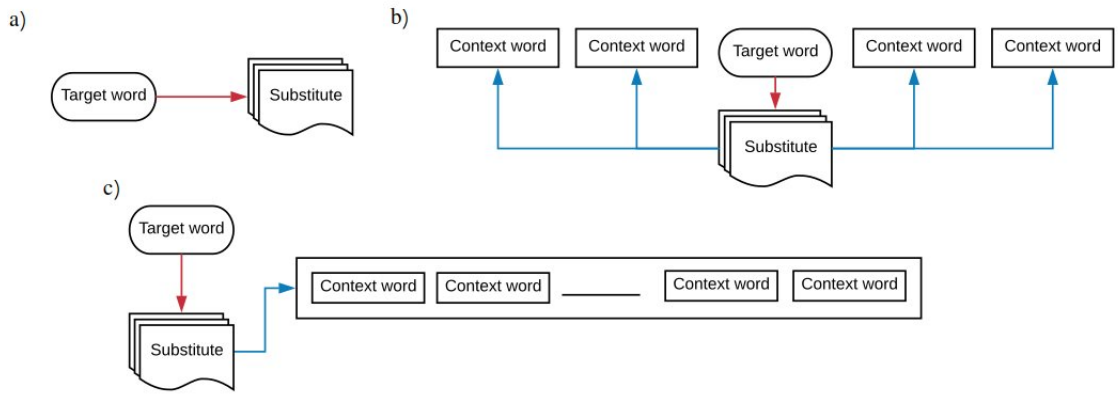

Figure 1: Illustration of the type of context information the different methods use: a) *tTs* uses target to substitute similarity only (Section 4.1); b) *AddCos* also uses similarities between a candidate and each of the words in the surrounding context (Section 4.2); c) *c2vf* makes use instead of a unique embedding representing the whole sentential context (Section 4.3).

context embeddings. The proposed measures differ in the way they combine the score elements together, using either an arithmetic or geometrical mean. We choose the more flexible additive approach which, contrary to the multiplicative variants, does not require high similarities in all elements of the product to highly rank a substitute, but can yield a high score even if one of the elements in the sum is zero. The *Add* measure (equation (1), hereafter called *AddCos* because of the Cosine function applied to the vector representations of words and contexts) estimates the substitutability of a candidate substitute s of the target word t in context C, where C corresponds to the set of the target word's context elements in the sentence, and c corresponds to an individual context element.

$$AddCos(t, s, C) = \frac{cos(s, t) + \sum_{c \in C} cos(s, c)}{|C| + 1} \quad (1)$$

The vectors used by the original method are syntax-based embeddings created with *word2vecf* (Levy and Goldberg, 2014). We use the lighter adaptation proposed by Apidianaki et al. (2018) which circumvents the need for syntactic analysis, and use 300-dimensional skip-gram word and context embeddings trained on the 4B words of the Annotated Gigaword corpus (Napoles et al., 2012).

We apply the AddCos method to ELMo as well as to FC-ELMo embeddings. When using standard ELMo embeddings, the target and context word representations of a sentence are their corresponding ELMo vector, and the vector of a candidate substitute is obtained by substituting the target word by the candidate in the sentence, as described in Section 4.1. To adapt this to FC-ELMo embeddings, substitute representations are replaced by their corresponding FC-ELMo vectors.

4.3 The *context2vec*-based model

The *context2vec* (*c2v*) model jointly learns context and word embeddings using bidirectional LSTM (Melamud et al., 2016). The proposed neural network is based on word2vec's CBOW architecture (Mikolov et al., 2013), but replaces its naive context modeling of averaged word embeddings in a fixed window with a full-sentence neural representation of context obtained using bidirectional LSTM. Words and contexts are embedded in the same space, which allows for calculating target-to-context ($t2c$), context-to-context ($c2c$) and target-to-target ($t2t$) similarities. A score for a candidate substitute is computed using the following formula:

$$\text{c2v\_score} = \frac{cos(s, t) + 1}{2} \times \frac{cos(s, C) + 1}{2} \quad (2)$$

where t and s are the word embeddings of the target and the substitute, and C is the *c2v* context vector of the sentence with an empty slot at the target's position. We use the 600-dimensional c2v embeddings released by Melamud et al. (2016).

We also use Equation (2) (hereafter called *c2vf*) with standard ELMo and FC-ELMo vectors. As with the AddCos method, we represent the target word in context by its ELMo embedding, and the substitute vectors are obtained with the in-place substitution approach described above (cf. Sections 4.1, 4.2). The context vector (C) is the average of the ELMo embeddings of all words in the context. To test FC-ELMo embeddings in this setting, each substitute is represented by its FC-ELMo embedding.

Finally, we experiment with FC-c2v embeddings, i.e. standard context2vec embeddings (Melamud et al., 2016) tuned on the FC dataset. Target and context are represented with standard c2v embeddings, and substitutes are represented with FC-c2v embeddings.

4.4 Baselines

We compare our models to two context-insensitive baselines that solely rely on the target-to-substitute similarity of standard, pre-trained word embeddings: 300-dimensional GloVe vectors (Pennington et al., 2014)[5] and 300-dimensional FastText vectors, both trained on Common Crawl (Mikolov et al., 2018).[6] Similar to tTs (Section 4.1), this approach only considers target-to-substitute similarity. With these uncontextualized embeddings the ranking proposed for each target word is always the same regardless of context.

We also propose an enriched version of the two baseline models by adding a simple representation of context consisting of the average of the embeddings of words in a sentence. We then compare target and substitute vectors to the generated context vector using the context2vec formula (Equation 2).

5 Evaluation

We compare the performance of the proposed models on a ranking task, where models assign scores to all candidate substitutes for a target word ($S = \{s_1, s_2, ..., s_n\}$) according to their suitability in new contexts. For evaluation, we use the dataset from the SemEval-2007 Lexical Substitution task (McCarthy and Navigli, 2007). The full dataset consists of 2,010 sentences, 10 for each of 201 target words (nouns, verbs, adjectives and adverbs), extracted from the English Internet Corpus (Sharoff, 2006), and annotated by five native English speakers. Words in this lexical sample were selected to ensure variety of senses. We filter the test set to preserve target words and substitutes present in PPDB 2.0 (XXL) and having a vector available in all tested models, to ensure all methods use exactly the same substitute pool per target word. Target words for which none or only one substitute was left were removed. The filtered test set used in our experiments includes 158 target words and 1,584 sentences.

The ranking performed by each model is compared to the gold ranking by means of Generalized Average Precision (GAP) (Kishida, 2005). GAP measures the quality of a ranking by comparing the resulting ranked list with the gold standard annotation, using substitution frequency as weights (i.e. number of annotators that suggested each substitute). GAP scores range between 0 and 1. A score of 1 indicates a perfect ranking where all correct substitutes precede all incorrect ones, and high-weight substitutes precede low-weight ones (Thater et al., 2010). We use the GAP implementation in Melamud et al. (2015)[7].

6 Results

The results of the proposed methods in the substitute ranking task are given in Table 3. The standard context2vec (c2v) model (Melamud et al., 2016) outperforms other methods, including those based on

[5] https://nlp.stanford.edu/projects/glove

[6] https://fasttext.cc/docs/en/english-vectors.html

[7] https://github.com/orenmel/lexsub

Method	Vectors	GAP
AddCos (c=1)	Skip-gram (Apidianaki et al., 2018)	0.527
	ELMo-avg	0.527
	ELMo-top	0.513
	FC-ELMo-avg	0.494
	FC-ELMo-top	0.491
AddCos (c=4)	Skip-gram (Apidianaki et al., 2018)	0.520
	ELMo-avg	0.498
	ELMo-top	0.476
	FC-ELMo-avg	0.481
	FC-ELMo-top	0.478
c2vf	UkWac c2v (Melamud et al., 2016)	**0.587**
	FC-c2v	0.492
	ELMo-avg	0.529
	ELMo-top	0.516
	FC-ELMo-avg	0.490
	FC-ELMo-top	0.480
tTs	ELMo-avg (Peters et al., 2018a)	0.534
	ELMo-top (Peters et al., 2018a)	0.531
	FC-ELMo-avg	0.493
	FC-ELMo-top	0.488
Glove + context	Glove (Pennington et al., 2014)	0.467
Fasttext + context	Fasttext (Mikolov et al., 2018)	0.491
Baselines	Glove (Pennington et al., 2014)	0.465
	Fasttext (Mikolov et al., 2018)	0.485

Table 3: Results of the substitute ranking experiment with all methods and embedding types. For AddCos models, *c* refers to the size of the window.

ELMo vectors. The superiority of context2vec is due to its training objective: context2vec is explicitly trained with pairs of target words and sentential contexts, optimizing the similarity of context vectors and potential fillers. This training objective makes the model highly suited for the LexSub task. In contrast, ELMo representations are trained as a general language model that predicts the immediate next tokens, while other types of similarity (e.g. target-to-substitute and substitute-to-context) used by the other methods are not explicitly accounted for. The underlying assumption of the AddCos and context2vec models that these similarities need to be high for good substitutes, does not thus apply in the case of ELMo embeddings.

The ELMo-avg and ELMo-top configurations – which use the top layer or an average of the three layers of the biLM – give comparable results, with ELMo-avg performing slightly better in all settings. Peters et al. (2018b) present a thorough analysis of the performance of different layers of the biLM models in different tasks, which shows that top layers are better suited for semantic-related tasks than lower layers. In the supervised word sense disambiguation (WSD) evaluation presented in Peters et al. (2018a) results obtained using the top layer were also slightly better than those of the middle layer. We believe the slight advantage of the ELMo-avg models, compared to ELMo-top, in LexSub, highlights an important difference between the two tasks. In LexSub, the selected substitute needs to correctly describe the meaning of the target word instance and to be a good fit in the context, producing a natural-sounding sentence. Substitute candidates for a word are often near-synonyms that would be preferred in different contexts. On the contrary, selection in WSD mainly relies on semantic adequacy. For example, when selecting one among available senses of a word in a resource like WordNet, the synonyms found in the selected synset might not all be good in-context substitutes. We believe the ELMo representation obtained by averaging the three layers to contain information regarding both the semantic and the syntactic

Sentence	*on the **way** out of the parking lot johnny felt a thump*
Candidate substitutes for *way.n*	sense, means, aspect, technique, passage, respect, direction, characteristic, journey, method, route, practice, fashion, manner
Gold ranking	route (3), passage (1), journey (1)

Table 4: A new instance of the target noun *way* (*way.n*) from the SemEval-2007 test set, the candidate substitutes extracted for the word from the PPDB XXL package, and the gold substitute ranking used for evaluation.

Method	Vectors	Ranked substitutes
c2vf	UkWac c2v (Melamud et al., 2016)	**route**, **journey**, manner, **passage**, direction, means, sense, aspect, method, fashion, respect, technique, characteristic, practice
tTs	ELMo-avg (Peters et al., 2018a)	**route, journey**, manner, direction, **passage**, method, means, respect, technique, sense, practice, aspect, fashion, characteristic
Baseline	Glove (Pennington et al., 2014)	sense, means, manner, **journey, route,** direction, respect, aspect, practice, method, technique, fashion, **passage**, characteristic
Baseline + ctxt	Glove (Pennington et al., 2014)	sense, means, manner, direction, respect, **journey**, aspect, **route**, practice, method, **passage**, technique, fashion, characteristic

Table 5: Examples of substitute rankings for the instance of the noun "*way*" given in Table 4 of the two best-performing methods (c2vf with standard c2v embeddings and tTs with ELMo-av embeddings) and the two methods with lowest GAP (baseline and baseline + context with Glove embeddings). Correct substitutes are marked in boldface to highlight their position in the ranking proposed by each model.

adequacy of a word. This does not contradict previous findings, since the semantics tasks in which the top ELMo layer was found to perform best were tasks that involve longer range dependencies and a more general notion of semantic similarity (e.g. coreference resolution).

The results obtained for FC-ELMO-* configurations show that ELMo representations do not benefit from the addition of discretized sense representations, rather the contrary. Whereas it looks like FC is introducing confusion to an already good model, we believe this could be due to the small amount of FC sentences used for tuning (100), which biases the model toward those sentences. Another reason could be that FC sentences selected using the PMI metric for a target-substitute pair are not always high quality, i.e. they might not contain, or not be representative enough, of the sense being expressed. In future work, we intend to experiment with a larger number of sentences for tuning, and with different ways for measuring the quality of sentences to be included in the FC resource.

The baseline methods that use uncontextualized word embeddings are not very far behind most FC-ELMo-* models. However, they do seem to slightly benefit from adding context. FastText vectors are trained with word2vec's CBOW architecture using position-dependent weighting, which results in richer context representations and is, we believe, the main reason of its advantage over Glove on this task.

Finally, we observe that, for the AddCos method, a smaller context window around the target word (c=1) is consistently slightly more effective than a bigger one (c=4). This suggests that the most relevant context clues for lexical substitution are found in the close vicinity of a target word.

In Tables 4 and 5, we give an example of a new target word instance and the substitute ranking proposed by some of the models. In Table 4, we also provide the candidate substitutes considered for the target word *way*, which are its paraphrases in PPDB XXL that are also present among the gold standard annotations for this word. Numbers in parentheses denote the number of annotators that proposed each

substitute. We observe that the stronger models which use the c2v formula with the standard context2vec vectors (trained on UkWac) or the tTs method with ELMo-avg rank substitutes better than the baseline models.

7 Conclusion

We analyzed the behavior of different word and context representations in an in-context substitute ranking task. The compared methods differ as to the type of similarity they consider between words (target-to-substitute) and contexts (substitute-to-context). We experiment with the standard representations released for each approach, and fine-tune them to the LexSub task using an automatically compiled collection of sentences representing target-substitute pairs. Our results show that models trained with a slot-filling objective that optimizes the inter-dependencies between candidate substitutes and context, like context2vec, are a better fit for the LexSub task than purely context-based models, like ELMo. This is because they encode target-to-substitute similarity and local context appropriately for this task, which ensures the semantic and syntactic adequacy of the selected substitutes. The importance of these two parameters is also highlighted in our experiments by the performance of different combinations of ELMo layers, which shows that the substitute ranking task involves both semantic (top-layer) and syntactic (lower-layer) information.

In its current form, tuning on the sentences of the FC dataset does not seem to help the models. In future work, we plan to improve the quality of the substitute-focused contexts, to ensure a better representation of the meaning of target-substitute pairs that would be beneficial for this task. A large-scale resource of this type will be highly useful for training neural models for lexical substitution.

8 Acknowledgements

We would like to thank the anonymous reviewers for their thoughtful and constructive comments. This work has been supported by the French National Research Agency under project ANR-16-CE33-0013; the Allen AI Key Scientific Challenges program; the Google PhD Fellowship; and DARPA under grant numbers FA8750-13-2-0017 (the DEFT program) and HR0011-15-C-0115 (the LORELEI program).

References

Apidianaki, M., G. Wisniewski, A. Cocos, and C. Callison-Burch (2018). Automated paraphrase lattice creation for HyTER machine translation evaluation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, New Orleans, Louisiana, pp. 480–485.

Bannard, C. and C. Callison-Burch (2005). Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pp. 597–604.

Flekova, L. and I. Gurevych (2016). Supersense embeddings: A unified model for supersense interpretation, prediction, and utilization. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Volume 1, pp. 2029–2041.

Ganitkevitch, J., B. Van Durme, and C. Callison-Burch (2013). PPDB: The Paraphrase Database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, Georgia, pp. 758–764.

Iacobacci, I., M. T. Pilehvar, and R. Navigli (2015). Sensembed: Learning sense embeddings for word and relational similarity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Volume 1, pp. 95–105.

Kishida, K. (2005). *Property of average precision and its generalization: An examination of evaluation indicator for information retrieval experiments*. Technical Report NII-2005-014E, National Institute of Informatics Tokyo, Japan.

Kremer, G., K. Erk, S. Padó, and S. Thater (2014). What Substitutes Tell Us - Analysis of an "All-Words" Lexical Substitution Corpus. In *Proceedings of EACL*, Gothenburg, Sweden, pp. 540–549.

Levy, O. and Y. Goldberg (2014). Dependency-based word embeddings. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, Volume 2, pp. 302–308.

McCarthy, D. and R. Navigli (2007). Semeval-2007 task 10: English lexical substitution task. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, Prague, Czech Republic, pp. 48–53.

Melamud, O., J. Goldberger, and I. Dagan (2016). context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, Berlin, Germany, pp. 51–61.

Melamud, O., O. Levy, and I. Dagan (2015). A simple word embedding model for lexical substitution. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, Denver, Colorado, pp. 1–7.

Mikolov, T., K. Chen, G. Corrado, and J. Dean (2013). Efficient estimation of word representations in vector space. In *Proceedings of the International Conference on Learning Representations*, Scottsdale, Arizona.

Mikolov, T., E. Grave, P. Bojanowski, C. Puhrsch, and A. Joulin (2018). Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan, pp. 52–55.

Napoles, C., M. Gormley, and B. Van Durme (2012). Annotated Gigaword. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-scale Knowledge Extraction*, pp. 95–100.

Pavlick, E., P. Rastogi, J. Ganitkevitch, B. Van Durme, and C. Callison-Burch (2015). PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of ACL/IJCNLP*, Beijing, China, pp. 425–430.

Pennington, J., R. Socher, and C. Manning (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, Doha, Qatar, pp. 1532–1543.

Peters, M. E., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee, and L. Zettlemoyer (2018a). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana, pp. 2227–2237.

Peters, M. E., M. Neumann, L. Zettlemoyer, and W.-t. Yih (2018b). Dissecting contextual word embeddings: Architecture and representation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium, pp. 1499–1509.

Rothe, S. and H. Schütze (2015). Autoextend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, Volume 1, pp. 1793–1803.

Sharoff, S. (2006). Open-source corpora: Using the net to fish for linguistic data. *International Journal of Corpus Linguistics 11*(4), 435–462.

Thater, S., H. Fürstenau, and M. Pinkal (2010). Contextualizing semantic representations using syntactically enriched vector models. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, Uppsala, Sweden, pp. 948–957.