ESC: Redesigning WSD with Extractive Sense Comprehension

Edoardo Barba[1] **Tommaso Pasini**[2,*] **Roberto Navigli**[1]
[1]Sapienza NLP Group, Department of Computer Science, Sapienza University of Rome
[2]Department of Computer Science, University of Copenhagen
{barba,navigli}@di.uniroma1.it
tommaso.pasini@di.ku.dk

Abstract

Word Sense Disambiguation (WSD) is a historical NLP task aimed at linking words in contexts to discrete sense inventories and it is usually cast as a multi-label classification task. Recently, several neural approaches have employed sense definitions to better represent word meanings. Yet, these approaches do not observe the input sentence and the sense definition candidates all at once, thus potentially reducing the model performance and generalization power. We cope with this issue by reframing WSD as a span extraction problem — which we called Extractive Sense Comprehension (ESC) — and propose ESCHER, a transformer-based neural architecture for this new formulation. By means of an extensive array of experiments, we show that ESC unleashes the full potential of our model, leading it to outdo all of its competitors and to set a new state of the art on the English WSD task. In the few-shot scenario, ESCHER proves to exploit training data efficiently, attaining the same performance as its closest competitor while relying on almost three times fewer annotations. Furthermore, ESCHER can nimbly combine data annotated with senses from different lexical resources, achieving performances that were previously out of everyone's reach. The model along with data is available at https://github.com/SapienzaNLP/esc.

1 Introduction

Being able to link a piece of raw text to a knowledge base is fundamental in NLP (Navigli, 2009; McCoy et al., 2019; Bender and Koller, 2020), as it can aid neural models to ground their representations on structured resources and enable Natural Language Understanding (Navigli, 2018). A task that is key to achieving this goal is Word Sense Disambiguation (WSD), where, given a sentence with a target word, a model has to predict its most suitable meaning from a predefined set of labels, i.e., its senses. WSD has not only considerably improved its performance with the advent of deep learning (by around 15 F1 points in 15 years), but it has also shown its benefits in downstream applications such as Neural Machine Translation (Liu et al., 2018; Pu et al., 2018) and Information Extraction (Moro and Navigli, 2013; Delli Bovi et al., 2015), while also being leveraged to enrich the contextual representations of neural models (Peters et al., 2019; Zhang et al., 2019). However, WSD has mostly been framed as a multi-label classification task (Raganato et al., 2017b; Hadiwinoto et al., 2019) over a very large vocabulary of discrete senses. This formulation may limit a model's capabilities to properly represent word meanings, as each sense is only defined by means of its occurrences in a training set, while its inherent meaning remains linguistically unexpressed. Furthermore, rare or unseen senses are either poorly modeled or cannot be modeled at all. These problems have recently been mitigated by integrating sense definitions (glosses) within neural architectures (Kumar et al., 2019; Huang et al., 2019; Blevins and Zettlemoyer, 2020). Yet, despite their large improvements, none of these models attends all the possible definitions of a target word at once, and therefore each lacks the ability to represent both the input context and the candidate definitions together.

Inspired by the Extractive Reading Comprehension framework (Rajpurkar et al., 2016) in the field of Question Answering (QA), we cope with these issues and reframe the WSD problem as a novel text extraction task, which we have called *Extractive Sense Comprehension* (ESC). In this setting, a model receives as input a sentence with a target word and all its possible sense definitions. Then, we request the model to extract the text span associated with the gloss expressing the target word's most suitable meaning. Within this frame-

*Work carried out while at the Sapienza University of Rome.

Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 4661–4672
June 6–11, 2021.

work, we also propose a transformer-based architecture (ESCHER) that implements the ESC task by attending to the input context and target word definitions jointly. Through an extensive experimental setting, we show that ESCHER surpasses former state-of-the-art approaches by a large margin while, at the same time, requiring almost 3 times less training data points to attain performances comparable to its strongest competitor in a few-shot setting. Furthermore, thanks to our new formulation, the proposed model can effectively carry out predictions across different sense repositories and combine distinct inventories with unmatched nimbleness, attaining even higher results than when limited to a single resource only.

To summarize, this paper brings the following novel contributions:

1. The Extractive Sense Comprehension task (ESC), i.e., a reframing of the Word Sense Disambiguation problem.

2. ESCHER: a transformer-based architecture for ESC, outperforming all the other modern architectures on the WSD task.

3. An extensive study of the proposed model in different training regimes, i.e., in 0-shot, few-shot and fully-supervised settings.

4. A study on combining data annotated with distinct lexicographic resources.

Besides its performance advantages, ESC also comes with other benefits: it does not require a large output vocabulary, and it eases the joint use of corpora annotated with different inventories.

2 Related Work

Word Sense Disambiguation (WSD) is one of the long-standing problems in lexical semantics, introduced for the first time in the context of Machine Translation by Weaver (1949). WSD aims at linking a word in context to its most suitable meaning in a predefined sense inventory, which is usually a dictionary where each entry defines a concept via a definition (gloss) and a set of examples. Most approaches to WSD rely on WordNet (Miller et al., 1990) as the underlying inventory of senses for the English language, and SemCor (Miller et al., 1993) as training corpus. WordNet organizes lexical-semantic information by means of a graph where sets of synonyms are grouped into synsets (concepts) and edges are typed semantic relations.

While early neural models used WordNet as a mere repository of senses (Raganato et al., 2017b; Hadiwinoto et al., 2019), more recent approaches have started to exploit sense definitions (Kumar et al., 2019; Blevins and Zettlemoyer, 2020) and relational information (Bevilacqua and Navigli, 2020; Conia and Navigli, 2021). Sense definitions, in particular, have been shown to be effective for modeling word senses (Luo et al., 2018; Kumar et al., 2019), as they provide information orthogonal to that available in the training data. This has been further investigated under different perspectives by Huang et al. (2019, GlossBERT), Blevins and Zettlemoyer (2020, BEM) and Bevilacqua et al. (2020, Generationary). GlossBERT casts the WSD problem as a binary classification task where, given a word in context and one of its dictionary definitions, it determines whether this definition matches the word meaning expressed in the context. BEM employs a bi-encoder to represent the target word and its sense definitions within the same space. Generationary, instead, has predefined sense inventories at its disposal and directly generates a definition given a word in its context. The strength of these approaches lies in the fact that glosses allow senses that are under-represented within the training corpus to be modeled, hence mitigating the long-standing paucity of sense-annotated data (Pasini, 2020). Nevertheless, none of the above approaches can exploit all definitions at once: indeed, glosses are either provided one at a time (GlossBERT), modeled with one vector only and independently from each other (BEM), or used individually as target text to be generated (Generationary).

Our new formulation (ESC) for the WSD problem stands out from previous approaches inasmuch as it is the first to access the input context and all the target word's definitions together, while, at the same time, dropping the requirement of a predefined sense inventory. Indeed, differently from its competitors, our proposed approach (ESCHER) can scale effectively across different lexical resources even when they were not available at the time of training.

3 Methodology

In what follows, we first formalize the Extractive Sense Comprehension task (Section 3.1), then in-

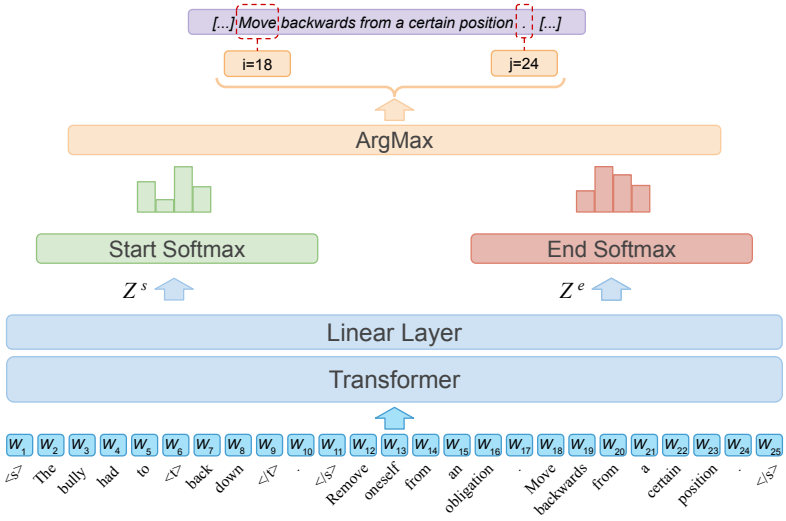


Figure 1: Depiction of ESCHER. The model takes as input a sentence concatenated with all the target word's definitions and outputs two indices indicating the start and end token in the input text of the target word definition.

troduce ESCHER, a transformer-based architecture for ESC (Section 3.2), and finally put forward a novel approach for mitigating the bias towards the most frequent meanings (Kilgarriff, 2004) within training data (Section 3.3).

3.1 Extractive Sense Comprehension

To unleash the full potential of attention-based models on the Word Sense Disambiguation task, we reframe WSD as a span-extraction problem. Formally, given a sense inventory S, we first define the definitional context $\mathcal{D}_{\hat{w}}$ for the target word $\hat{w}$ as the concatenation of all the possible definitions $d_1, \ldots, d_k$ in S for $\hat{w}$, i.e., $\mathcal{D}_{\hat{w}} = w_1^{d_1} \ldots w_{|d_1|}^{d_1} \ldots w_1^{d_k} \ldots w_{|d_k|}^{d_k}$, where $w_i^{d_z}$ is the i-th word of the gloss d_z ($1 \leq z \leq k$). Then, we reformulate the task as follows: given a target word $\hat{w}$, a context c in which $\hat{w}$ occurs and the definitional context $\mathcal{D}_{\hat{w}}$, a model has to find the interval $[i^*, j^*]$ in $\mathcal{D}_{\hat{w}}$ which identifies the correct definition $d^* \in \mathcal{D}_{\hat{w}}$ of $\hat{w}$ in c. This formulation, on the one hand, aids to better characterize word meanings, thanks to the inclusion of all the target word definitions as additional input. On the other hand, it also relieves the burden of a large output vocabulary – typically in the order of tens of thousands of meanings – which makes the classification cumbersome.

3.2 ESCHER

We now introduce a transformer-based model for the ESC task (Figure 1). It takes as input a context c with a target word $\hat{w}$[1] concatenated with $\mathcal{D}_{\hat{w}}$. The target word $\hat{w}$ is surrounded by the tags $<t>$ and $</t>$ and each definition in $\mathcal{D}_{\hat{w}}$ has the first letter capitalized and a period at the end. We separate the context c and the definitional context $\mathcal{D}_{\hat{w}}$ with the special symbol $</s>$ and surround the whole text with the tags $<s>$ and $</s>$.[2]

Formally, given the input:

$$m = <s> w_1 \ldots <t> \hat{w} </t> \ldots w_n </s>$$
$$w_1^{d_1} \ldots w_{|d_1|}^{d_1} \ldots w_1^{d_k} \ldots w_{|d_k|}^{d_k} </s>$$

of length l, the model computes the span (i, j) containing the predicted gloss for the target word $\hat{w}$ as follows:

$$H = \texttt{transformer}(m)$$
$$Z = W^T H + b$$
$$Z^s = \begin{bmatrix} Z_{11} & \ldots & Z_{1l} \end{bmatrix}$$
$$Z^e = \begin{bmatrix} Z_{21} & \ldots & Z_{2l} \end{bmatrix}$$

where `transformer` can be any transformer-based architecture, $H \in \mathbb{R}^{f \times l}$ is the matrix of hidden states,[3] and $W \in \mathbb{R}^{f \times 2}$ and $b \in \mathbb{R}^2$ are trainable parameters. Z^s and Z^e are two variables containing the logits for each word w_u indicating, respectively, whether it is the start or the end of the correct definition for target word $\hat{w}$.

[1] For the sake of simplicity, in the following we use *word* to refer to subwords, words and multiwords.

[2] The $<s>$ and $</s>$ tags can be any special token in a model vocabulary that have been used to divide texts at pretraining time.

[3] f indicates the number of dimensions of each hidden state.

Finally, we train the model by averaging two distinct cross-entropy losses that we compute for the start and end indices:

$$\mathcal{L}_s = -Z^s_{i^*} + log \sum_{v=1}^{l} exp(Z^s_v)$$

$$\mathcal{L}_e = -Z^e_{j^*} + log \sum_{v=1}^{l} exp(Z^e_v)$$

where $Z^s_{i^*}$ and $Z^e_{j^*}$ are the scores associated with the correct start and end indices.

At prediction time, rather than allowing the system to output a span that does not correspond precisely to any definition in $\mathcal{D}_{\hat{w}}$, the model outputs a pair (i, j) such that a definition $d_k \in \mathcal{D}_{\hat{w}}$ starts in i and ends in j and its probability is the maximum across all the other gloss spans in $\mathcal{D}_{\hat{w}}$. Formally, the model selects its output as follows:

$$\texttt{output} = \arg\max_{(i,j)} P(w_i, w_j)$$

$$P(w_i, w_j) = P(w_i = start \mid Z^s) \times P(w_j = end \mid Z^e)$$

$$P(w_u = start \mid Z^s) = \frac{exp(Z^s_u)}{\sum_{v=1}^{l} exp(Z^s_v)}$$

$$P(w_u = end \mid Z^e) = \frac{exp(Z^e_u)}{\sum_{v=1}^{l} exp(Z^e_v)}$$

where $P(w_u = start \mid Z^s)$ and $P(w_u = end \mid Z^e)$ indicate the probability that w_u is the start or the end of any of the k definitions, respectively.

3.3 Rebalancing the Most Frequent Sense Bias

While our approach already allows all the possible definitions of a word to be contextualized by jointly encoding them together with the context sentence, it may still suffer from the high unbalance in sense distribution (Kilgarriff, 2004) and be biased towards the most frequent definition regardless of its contextualization. Our framework allows this issue to be dealt with in an elegant way, which we have called Gloss Noise (GN). GN counterbalances this bias by lowering the prior probability of the most frequent glosses. That is, inspired by the negative sampling technique (Mikolov et al., 2013), GN adds, to each training example, k frequent definitions that are not related to the target word. We sample the k glosses from the following multinomial distribution:

$$p(d_i) = \frac{f_{d_i}}{\sum_{j=1}^{|D|} f_{d_j}}$$

where D is the set of all possible definitions in the training set and f_{d_i} is the frequency of the i-th definition in a sense-tagged corpus. The value of k, instead, is sampled from a Poisson distribution with $\lambda = 1$, so that the expected number of added definitions is equal to 1. This allows the discrepancy between the training and prediction phases to be kept as small as possible, while also introducing negative signals for frequent senses. Indeed, Gloss Noise ensures that the expected number of times a definition is added as a negative example is equal to the number of times it is seen as a correct one, thereby counterbalancing the high rate at which frequent definitions are seen only as positive examples without overly affecting rare senses.

4 Standard WSD Evaluation

In this Section we introduce the experimental setting we use to evaluate the proposed framework and neural architecture.

4.1 Setup

Data We use the evaluation suite made available by Raganato et al. (2017a) for the English Word Sense Disambiguation task. It includes SemCor (Miller et al., 1993) for training, i.e., a corpus containing 33,362 sentences and 226,036 instances annotated manually with senses from WordNet 3.0. As common practice, we use SemEval-2007 (**SE07**; Pradhan et al., 2007) as development set. For testing, we consider all the remaining datasets in the suite, i.e., Senseval-2 (**SE2**; Edmonds and Cotton, 2001), Senseval-3 (**SE3**; Snyder and Palmer, 2004), SemEval-2013 (**SE13**; Navigli et al., 2013), SemEval-2015 (**SE15**; Moro and Navigli, 2015) and their concatenation (ALL).[4]

In order to measure the extent to which systems generalize to rare and unseen words and definitions (zero-shot settings), we also consider five other test sets that we created from the ALL dataset:

i) **MFS**, which contains test instances tagged with the most frequent sense for the target word in the training set;

ii) **LFS**, which contains test instances that are tagged with a sense that is not the most frequent for the target word and that was seen at least once during training;

[4] We note that the evaluation suite includes the dev set, i.e., SemEval-2007, within the ALL dataset, and so do we.

iii) **0-lex**, which contains test instances whose lexeme[5] was never seen as a target word during training;

iv) **0-lex-def**,[6] which contains test instances with a definition that was never seen associated with the target lexeme during training;

v) **0-def**, which contains test instances whose definition has never been seen during training. We note that 0-def differs from 0-lex-def as a definition is tied in WordNet to a synset, i.e., a set of synonymous senses, rather than to a sense; therefore the same definition may be seen associated with different lexemes.

Comparison Systems As baselines, we consider the Most Frequent Sense computed on the training set (MFS SemCor) and two neural models featuring BERT$_{large}$ and BART$_{large}$ as text encoders, with a linear classifier over the whole sense vocabulary on top. As for the BERT$_{large}$ baseline, we follow Blevins and Zettlemoyer (2020) and keep BERT$_{large}$ weights fixed, while for BART$_{large}$ we finetune the whole model.

As competitors, we consider the following models: GLU (Hadiwinoto et al., 2019), which keeps BERT weights frozen and trains a gated linear unit on top of it; SVC[7] (Vial et al., 2019), which uses a vocabulary compression technique; EWISE (Kumar et al., 2019); GlossBERT (Huang et al., 2019); BEM[8] (Blevins and Zettlemoyer, 2020) and EWISER (Bevilacqua and Navigli, 2020), which take advantage of external knowledge such as glosses and semantic relations. We note that EWISER uses a different development set, hence its results are not fully comparable with the others. Finally, we also consider two nearest-neighbour approaches based on synset embedding and vector similarity, i.e., LMMS (Loureiro and Jorge, 2019) and ARES (Scarlini et al., 2020).

ESCHER Setting We use BART$_{large}$ (Lewis et al., 2020; Wolf et al., 2020) as transformer architecture[9] owing to the fact that it is among the strongest models on reading comprehension tasks such as SQuAD (Rajpurkar et al., 2016) and it allows us to feed sequences up to 1024 subtokens long.[10] We use the output of its last decoder layer to represent the input tokens and compute the start and end token distributions. We note that ESCHER is directly comparable to the BART$_{large}$ baseline in terms of model complexity as both use the same transformer model with one linear layer on top.

We finetune the whole ESCHER architecture with the Rectified Adam (Liu et al., 2020) optimizer with learning rate set to $1 \cdot e^{-5}$ for up to 300,000 steps, 20 steps of gradient accumulation and batches made of 700 tokens.[11] In what follows, we report the results for our model with and without Gloss Noise (Section 3.3), denoting them as ESCHER and ESCHER$_{\text{No-GN}}$, respectively.

4.2 Results

Framework Benchmark In Table 1 we report the F1 scores of ESCHER, ESCHER$_{\text{No-GN}}$ and all the other systems. By comparing BART$_{large}$ and ESCHER, we can measure the effectiveness of our proposed framework, ESC, on the performance of a transformer-based architecture. Indeed, the two architectures are nearly identical except for the last layer, where, for each token, BART$_{large}$ makes a prediction across the whole sense vocabulary, while ESCHER performs a binary classification. Thus, the large difference between the two models (8.5 F1 points) suggests that the Extractive Sense Comprehension formulation of WSD allows the potential of transformer-based architectures to be fully exploited, and, therefore, attain better performance.

When Gloss Noise is enabled (ESCHER row), our model gains 1 F1 point in comparison to when it is disabled (ESCHER$_{\text{No-GN}}$). This highlights that directly mitigating the bias towards the Most Frequent Senses during training is fundamental to making our approach as effective as possible.

Finally, thanks to our new formulation of the WSD problem, a simple model such as ESCHER outperforms all the other approaches by a large margin on the ALL dataset, beating the previous state of the art by 1.7 points (BEM). This corroborates our hunch that the Extractive Sense Comprehension task is an extremely effective formulation of WSD for transformer-based architectures.

[5] A (lemma, part of speech) pair.

[6] We identify a sense as a pair (lexeme, definition).

[7] Similarly to Blevins and Zettlemoyer (2020), we report the best results of the SVC single model trained on SemCor only.

[8] BEM is the state-of-the-art model in this setting at the time of writing.

[9] Please see Appendix A for experiments with different transformer pretrained models.

[10] As for the $<t>$ and $</t>$ symbols, we used the words $<classify>$ and $</classify>$.

[11] Please refer to Appendix B for further details on the training.

	Model	Dev Set	Test Sets				Concatenation of all Datasets				
		SE07	SE2	SE3	SE13	SE15	Nouns	Verbs	Adj.	Adv.	ALL
Baselines	MFS SemCor	54.5	65.6	66.0	63.8	67.1	67.7	49.8	73.1	80.5	65.5
	BERT$_{base}$	68.6	75.9	74.4	70.6	75.2	75.7	63.7	78.0	85.8	73.7
	BART$_{large}$	63.5	75.0	72.2	69.3	74.2	74.0	61.6	76.9	86.1	72.2
Prior work	EWISE‡	67.3	73.8	71.1	69.4	74.5	74.0	60.2	78.0	82.1	71.8
	GLU	68.1	75.5	73.6	71.1	76.2	—	—	—	—	74.1
	LMMS††	68.1	76.3	75.6	75.1	77.0	—	—	—	—	75.4
	SVC	—	—	—	—	—	—	—	—	—	75.6
	GlossBERT†	72.5	77.7	75.2	76.1	80.4	79.8	67.1	79.6	87.4	77.0
	ARES††	71.0	78.0	77.1	78.7	75.0	80.6	68.3	80.5	83.5	77.9
	EWISER‡	71.0	78.9	**78.4**	78.9	79.3	81.7	66.3	81.2	85.8	78.3
	BEM†	74.5	79.4	77.4	79.7	81.7	81.4	68.5	83.0	**87.9**	79.0
Ours	ESCHER$_{\text{No-GN}}^{\dagger}$	75.0	80.5	76.9	81.1	83.0	83.0	68.5	81.9	86.1	79.7
	ESCHER†	**76.3**	**81.7**	77.8	**82.2**	**83.2**	**83.9**	**69.3**	**83.8**	86.7	**80.7**

Table 1: Comparison of F1 scores for the all-words WSD task. EWISER uses SemEval-2015 (SE15) as development set. † indicates systems having access to sense definitions. ‡ indicates systems having access to sense definitions utilizing synset embeddings. †† indicates systems using a nearest-neighbor approach based on synset embeddings. The difference in performance attained on the ALL test between ESCHER and BEM (underlined) is statistically significant with $p \ll 0.01$ according to the McNemar's test (Dietterich, 1998).

Results on Rare and Unseen Senses In Table 2 we report the results of the three best-performing models, i.e., ESCHER, ESCHER$_{\text{No-GN}}$ and BEM, on five datasets, measuring how well models perform when dealing with rare words and meanings in different situations (cf. Section 4.1). ESCHER$_{\text{No-GN}}$ manages to outperform BEM on most datasets, hence already demonstrating that our new framing allows transformers to better generalize on rare words and senses. When enabling Gloss Noise, ESCHER achieves even higher performance on all datasets, falling behind BEM only on the MFS dataset. Interestingly enough, the comparison with BEM on the 0-lex-def and 0-def datasets shows that ESCHER can easily predict definitions that were either seen associated only with lexemes different from the input ones or not seen at all, while, in direct contrast, BEM performs poorly in both scenarios. A similar pattern is observed for the Least Frequent Senses (LFS) dataset, where ESCHER outperforms BEM by 3.6 F1 points at the cost of only 1 point less in predicting the most frequent meanings.

Model	MFS	LFS	0-lex	0-lex-def	0-def
BEM	**94.7**	52.1	91.2	67.1	68.2
ESCHER$_{\text{No-GN}}$	93.7	52.8	94.5	74.3	76.4
ESCHER	93.7	**55.7**	**95.1**	**75.0**	**76.8**

Table 2: Comparison of ESCHER against its competitors on MFS, LFS and zero-shot datasets.

5 Merging Multiple Knowledge Bases

Being able to combine datasets tagged with different inventories is a desirable ability for a model. Indeed, being able to use different datasets grants access to a larger number of examples, while, at the same time, removing the necessity of having one system for each inventory. However, merging distinct lexicographic resources is not a straightforward task and requires its own complex pipeline. An easier approach could be to concatenate datasets tagged with different vocabularies, which, nonetheless, would expose models to possibly different definitions for nearly identical meanings and to different levels of sense granularity. In this Section we therefore investigate the ability of ESCHER to manage data annotated with distinct sense inventories when simply joining them. To this end, we train ESCHER on the concatenation of SemCor and the Oxford Dictionary dataset (Chang et al., 2018) and compare its performance with the state-of-the-art system at the moment of writing, i.e. BEM, when trained on the same corpus.

5.1 The Oxford Dictionary Dataset

Chang et al. (2018) introduced a dataset containing roughly 785,000 instances for as many sentences and covering 79,004 senses of the Oxford Dictionary of English. The dataset is split

	Dataset	Polysemy	Exp. Polysemy	#Senses	#Instances
WordNet	SemCor	6.88	0.76	33,362	226,036
	SE07	8.48	0.29	375	455
	ALL	5.87	0.54	3,669	7,611
Oxford	Oxford$_{train}$	3.81	0.98	79,105	555,695
	Oxford$_{dev}$	6.69	0.68	33,197	78,550
	Oxford$_{test}$	6.79	0.76	37,714	151,306

Table 3: Statistics for training, development and test corpora annotated with two inventories: WordNet (top) and Oxford (bottom).

into `train` (Oxford$_{train}$), `dev` (Oxford$_{dev}$) and `test`[12] (Oxford$_{test}$). In Table 3 we report its statistics together with those of the training (SemCor), development (SE07) and test (ALL) sets of the standard evaluation suite. Specifically, we show the average polysemy of each dataset (Polysemy), the expressed polysemy (Exp. Polysemy), i.e., for each lexeme we compute the number of senses that appear in the dataset over the number of possible senses it can assume in the reference vocabulary and we average across all lexemes, the number of distinct senses (#Senses) and the number of instances (#Instances). As one can see, Oxford$_{train}$ contains more than two times the instances and senses of SemCor, while having roughly half of SemCor's polysemy but a higher expressed polysemy. As for Oxford$_{test}$, it contains a larger number of instances than ALL, and also a higher polysemy and expressed polysemy.

5.2 Setup

We analyze three different scenarios: i) **Standard**, where the system is trained on the same inventory with which it is tested, e.g., trained on Oxford$_{train}$ and tested on Oxford$_{test}$; ii) **Zero-shot**, in which the system is trained on one sense inventory and tested on the other, e.g,. trained on SemCor and tested on Oxford$_{test}$; and iii) **Joint**, in which the system is jointly trained with the two sense inventories. In order to combine the two different inventories, we train the model by alternating the batches made up of either SemCor or Oxford$_{train}$ instances. Since the number of instances in SemCor is lower than that in Oxford$_{train}$, we oversample SemCor by repeating its instances. Finally, we select the model with the best macro F1 averaged on the two validation datasets (SE07 and Oxford$_{dev}$). We add the subscript S, OT and $S+OT$ to models trained on SemCor, Oxford$_{train}$ and their concatenation, respectively.

[12] We refer to the one named `test_easy` in the original paper.

Model	SE07	ALL	OX$_{dev}$	OX$_{test}$
BEM$_S$	74.5	79.0	61.5	61.7
ESCHER$_S$	76.3	80.7	67.6	67.9
BEM$_{OT}$	56.9	67.2	84.2	84.3
ESCHER$_{OT}$	60.7	70.3	86.3	86.3
BEM$_{S+OT}$	74.9	78.8	85.0	85.2
ESCHER$_{S+OT}$	**77.8**	**81.5**	**87.6**	**87.7**

Table 4: Comparison of ESCHER and BEM when using different training sets, i.e., SemCor (BEM$_S$ and ESCHER$_S$), Oxford$_{train}$ (BEM$_{OT}$ and ESCHER$_{OT}$) and their concatenation (BEM$_{s+OT}$ and ESCHER$_{S+OT}$).

5.3 Results

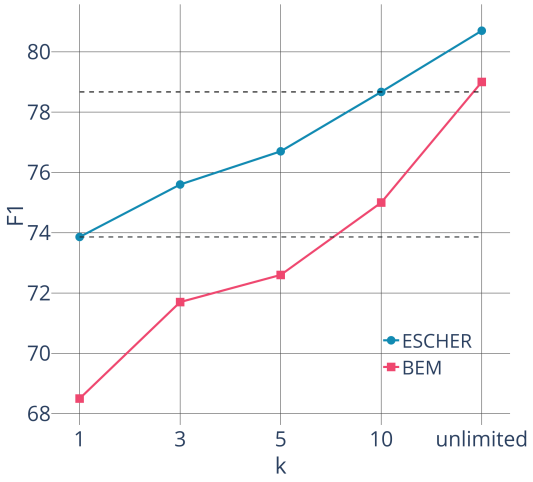
As one can see from Table 4, ESCHER outperforms BEM in all settings. That is, when trained with one inventory and tested on a dataset tagged with the other inventory (BEM$_S$ and ESCHER$_S$ on the Oxford$_{test}$ and BEM$_{OT}$ and ESCHER$_{OT}$ on ALL), ESCHER attains 6 and 3 points higher performance, respectively, than its competitor. This result is not important per se, but it also suggests that ESC does not bind the model to a single lexical knowledge base. Indeed, by extensively leveraging sense definitions, it allows a transformer-based model to scale on multiple inventories as long as they provide at least one definition for each meaning. BEM, instead, by encoding each gloss independently, falls short in representing definitions that were previously unseen, as also shown in Section 4.2.

When trained on SemCor and Oxford$_{train}$ together, not only can ESCHER handle the two inventories that coexist in the training set effectively, but it also leverages them at its own convenience, achieving 81.5 F1 points on ALL, in contrast to BEM which performs slightly worse than when trained in the Standard scenario.

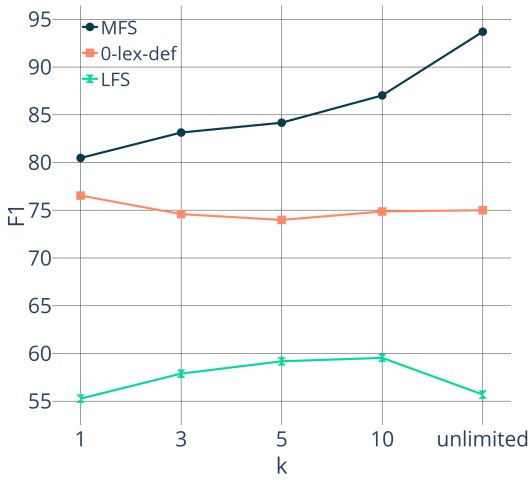
6 Few-Shot Evaluation

We now move to analyzing the performances of ESCHER in a few-shot scenario, i.e., when the number of samples available for each sense is limited.

Setting We compare ESCHER against BEM, and report the F1 scores on the ALL dataset when varying the number k of training instances per sense in $\{1, 3, 5, 10, unlimited\}$. We show in Table 5 the



(a) ALL dataset.



(b) LFS and 0-lex-def datasets.

Figure 2: (a) F1 Performances of BEM and ESCHER on the ALL evaluation dataset, when varying the number of instances per sense seen during training. (b) Performance of ESCHER on the MFS, LFS and 0-lex-def datasets, when varying the number of instances per sense seen during training.

k	Instances
1	33,206
3	64,814
5	83,068
10	109,751
unlimited	226,036

Table 5: Number of instances in the training set at different values of k.

number of instances drawn from SemCor that are seen at training time for each k.

We also report the F1 scores of ESCHER on the MFS, LFS and 0-lex-def datasets in the same scenario in order to investigate the extent to which the difference in the number of occurrences for each sense impacts the ability of the model to generalize on rare senses.

Results As one can see from Figure 2a,[13] ESCHER makes much more efficient use of training data than BEM, needing roughly one third of the instances to attain the same results. In fact, BEM needs more than 5 instances per sense (83,068 instances) to reach the same performance (73.9 F1 points) as that of ESCHER trained with $k = 1$ (33,206 instances). Furthermore, with roughly half of the instances ($k = 10$) ESCHER attains results that are in the same ballpark as the current state of the art. Interestingly enough, by looking at Figure 2b, we see that ESCHER's accuracy on the MFS instances rises when adding more examples. This is due to the fact that frequent senses get increasingly represented within the training set, therefore better matching the sense distribution in the test set. Similarly, the performance on the Least Frequent Senses also rises from $k = 1$ to $k = 10$, but slightly drops when considering the whole dataset. By manually inspecting the data we notice that this happens because most of the instances added to the dataset with $k = 10$ are tagged with the most frequent sense, therefore drastically skewing the sense distribution. Finally, the performance on 0-lex-def remains stable for all k, hence showing that, despite increasingly skewing the distribution towards the most frequent definitions, our approach can still provide meaningful representations for unseen senses.[14]

7 Error Analysis

In order to get a clear picture of the model's pitfalls and gain insights into possible directions for future work, we perform an analysis of ESCHER misclassifications on the ALL dataset. We find that the mistaken predictions belong to three main categories: most frequent sense bias, insufficient context and WordNet sense granularity. Since we already discussed the first of these in the previous sections, we focus here on the latter two.

[13] BEM chart from the original paper.

[14] We recall from Section 4.1 that we indicate as a sense a pair (lexeme, definition).

Insufficient Context Annotators often compiled the WSD evaluation datasets by considering each instance in the context of the documents they appear in. In contrast, WSD models typically take into account only the sentence surrounding the target word, discarding a large portion of the available context. This behavior causes a discrepancy where sentences do not provide enough information to disambiguate the target words therein. Indeed, ESCHER mistakes most often appear in sentences with an average length of 27 tokens, i.e., roughly 5 tokens less than the average length in ALL (32).

This suggests that moving the disambiguation context from sentences to documents may improve the performances of models as long as they are capable of handling longer sequences.

WordNet Sense Granularity The granularity of WordNet senses has been considered one of the main reasons behind the complexity of the WSD task (Palmer et al., 2007). To measure the extent to which this affects ESCHER's performance, we utilize the 45 domain-based labels introduced by Lacerra et al. (2020, CSI), which define macro categories for each WordNet sense. For instance, in the CSI inventory, the sense *argument%1:10:03::* belongs to the following domains: `Culture Anthropology and Society`, `Language and Linguistics` and `Communication and Telecomunication`.

To better understand the relation between ESCHER predictions and the gold annotations, for each misclassified instance in ALL, we compute the average Jaccard similarity between the CSI labels assigned to the gold annotation of that instance and those assigned to the sense predicted by ESCHER. As an example, ESCHER misclassified an instance annotated with the sense *argument%1:10:03::*, assigning to it the sense *argument%1:10:00::*. Examining the domains to which the predicted sense belongs, we can see a considerable overlap (and consequently a high Jaccard similarity) with the domains of the gold sense (i.e. *argument%1:10:03::*): `Culture Anthropology and Society`, `Politics Government and Nobility`, `Language and Linguistics` and `Communication and Telecomunication`.

As a term of comparison, we repeat the same procedure when considering a random baseline as WSD model, i.e., one that predicts for each instance a random sense among those of the target word. We find that ESCHER predictions have an average Jaccard similarity with the gold predictions of 0.49, whereas the random baseline achieves 0.27. This suggests that, even when providing a formally mistaken output, ESCHER still predicts a sense that is correlated, according to CSI labels, to the gold sense. Our analysis calls for further work to improve evaluation in WSD as the F1 score cannot discriminate between predictions that are clearly wrong and predictions that are just slightly different from the gold sense.

8 Conclusion

In this paper, we introduced a novel framing for the Word Sense Disambiguation problem inspired by the Extractive Reading Comprehension task in QA: given a word in a sentence and a text containing all its possible definitions, a model has to identify the span containing the correct definition for the target word. For this new formulation — which we called Extractive Sense Comprehension (ESC) — we devised a transformer-based architecture (ESCHER), which, differently from previous approaches, can look at all the target word definitions at once, alongside the input sentence. ESCHER surpasses the current state of the art by 1.7 points on the standard English all-words WSD task, thanks to its more efficient use of the training data. Also, when provided with only a few examples for each sense, ESCHER attains remarkable levels of performance, requiring roughly three times less annotated instances than its direct competitor to reach the same performances. Furthermore, our new formulation allows ESCHER to scale across different inventories and to combine them effectively. Indeed, when provided with data annotated with multiple vocabularies, it achieves even better results than when limited to one inventory only, with results in the 86-88% range.

As future work we plan to expand this framework so as to condition the prediction not only on the target word context and definitions, but also on the possible senses of its surrounding words.

The pretrained model, along with code and data, is available at `https://github.com/SapienzaNLP/esc`.

Acknowledgments

The authors gratefully acknowledge the support of the ERC Consolidator Grant MOUSSE No. 726487.

This work was supported in part by the MIUR under grant "Dipartimenti di eccellenza 2018-2022" of the Department of Computer Science of Sapienza University.

References

Emily M. Bender and Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5185–5198, Online. Association for Computational Linguistics.

Michele Bevilacqua, Marco Maru, and Roberto Navigli. 2020. Generationary or: "how we went beyond word sense inventories and learned to gloss". In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7207–7221, Online. Association for Computational Linguistics.

Michele Bevilacqua and Roberto Navigli. 2020. Breaking through the 80% glass ceiling: Raising the state of the art in word sense disambiguation by incorporating knowledge graph information. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2854–2864, Online. Association for Computational Linguistics.

Terra Blevins and Luke Zettlemoyer. 2020. Moving down the long tail of word sense disambiguation with gloss informed bi-encoders. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1006–1017, Online. Association for Computational Linguistics.

Ting-Yun Chang, Ta-Chung Chi, Shang-Chi Tsai, and Yun-Nung Chen. 2018. xsense: Learning sense-separated sparse representations and textual definitions for explainable word sense networks. *arXiv preprint arXiv:1809.03348*.

Simone Conia and Roberto Navigli. 2021. Framing Word Sense Disambiguation as a Multi-Label Problem for Model-Agnostic Knowledge Integration. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics.

Claudio Delli Bovi, Luca Telesca, and Roberto Navigli. 2015. Large-scale information extraction from textual definitions through deep syntactic and semantic analysis. *Transactions of the Association for Computational Linguistics*, 3:529–543.

Thomas G Dietterich. 1998. Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7):1895–1923.

Philip Edmonds and Scott Cotton. 2001. SENSEVAL-2: Overview. In *Proceedings of SENSEVAL-2 Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 1–5, Toulouse, France. Association for Computational Linguistics.

Christian Hadiwinoto, Hwee Tou Ng, and Wee Chung Gan. 2019. Improved word sense disambiguation using pre-trained contextualized word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5297–5306, Hong Kong, China. Association for Computational Linguistics.

Luyao Huang, Chi Sun, Xipeng Qiu, and Xuanjing Huang. 2019. GlossBERT: BERT for word sense disambiguation with gloss knowledge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3509–3514, Hong Kong, China. Association for Computational Linguistics.

Adam Kilgarriff. 2004. How dominant is the commonest sense of a word? In *Text, Speech and Dialogue*, pages 103–111, Berlin, Heidelberg. Springer Berlin Heidelberg.

Sawan Kumar, Sharmistha Jat, Karan Saxena, and Partha Talukdar. 2019. Zero-shot word sense disambiguation using sense definition embeddings. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5670–5681, Florence, Italy. Association for Computational Linguistics.

Caterina Lacerra, Michele Bevilacqua, Tommaso Pasini, and Roberto Navigli. 2020. CSI: A coarse sense inventory for 85% word sense disambiguation. In *Proceedings of the 34th Conference on Artificial Intelligence*, pages 8123–8130. AAAI Press.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Frederick Liu, Han Lu, and Graham Neubig. 2018. Handling homographs in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1336–1345, New Orleans, Louisiana. Association for Computational Linguistics.

Liyuan Liu, Haoming Jiang, Pengcheng He, Weizhu Chen, Xiaodong Liu, Jianfeng Gao, and Jiawei Han. 2020. On the variance of the adaptive learning rate

and beyond. In *International Conference on Learning Representations*.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A robustly optimized BERT pretraining approach. *arXiv preprint arXiv:1907.11692*.

Daniel Loureiro and Alípio Jorge. 2019. Language modelling makes sense: Propagating representations through WordNet for full-coverage word sense disambiguation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5682–5691, Florence, Italy. Association for Computational Linguistics.

Fuli Luo, Tianyu Liu, Qiaolin Xia, Baobao Chang, and Zhifang Sui. 2018. Incorporating glosses into neural word sense disambiguation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2473–2482, Melbourne, Australia. Association for Computational Linguistics.

Tom McCoy, Ellie Pavlick, and Tal Linzen. 2019. Right for the wrong reasons: Diagnosing syntactic heuristics in natural language inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3428–3448, Florence, Italy. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, G.s Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. *Advances in Neural Information Processing Systems*, 26:3111–3119.

George A. Miller, R.T. Beckwith, Christiane D. Fellbaum, D. Gross, and K. Miller. 1990. Introduction to WordNet: an online lexical database. *International Journal of Lexicography*, 3(4):235–244.

George A Miller, Claudia Leacock, Randee Tengi, and Ross T Bunker. 1993. A semantic concordance. In *Proceedings of the workshop on Human Language Technology*, pages 303–308. Association for Computational Linguistics.

Andrea Moro and Roberto Navigli. 2013. Integrating syntactic and semantic analysis into the open information extraction paradigm. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 2148–2154. IJCAI/AAAI.

Andrea Moro and Roberto Navigli. 2015. SemEval-2015 task 13: Multilingual all-words sense disambiguation and entity linking. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 288–297, Denver, Colorado. Association for Computational Linguistics.

Roberto Navigli. 2009. Word Sense Disambiguation: A survey. *ACM computing surveys (CSUR)*, 41(2):1–69.

Roberto Navigli. 2018. Natural Language Understanding: Instructions for (present and future) use. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5697–5702.

Roberto Navigli, David Jurgens, and Daniele Vannella. 2013. SemEval-2013 task 12: Multilingual word sense disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 222–231, Atlanta, Georgia, USA. Association for Computational Linguistics.

Martha Palmer, Hoa Trang Dang, and Christiane Fellbaum. 2007. Making fine-grained and coarse-grained sense distinctions, both manually and automatically. *Nat. Lang. Eng.*, 13(2):137–163.

Tommaso Pasini. 2020. The knowledge acquisition bottleneck problem in multilingual word sense disambiguation. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 4936–4942. International Joint Conferences on Artificial Intelligence Organization. Survey track.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 43–54, Hong Kong, China. Association for Computational Linguistics.

Sameer Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. 2007. SemEval-2007 task-17: English lexical sample, SRL and all words. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, pages 87–92, Prague, Czech Republic. Association for Computational Linguistics.

Xiao Pu, Nikolaos Pappas, James Henderson, and Andrei Popescu-Belis. 2018. Integrating weakly supervised word sense disambiguation into neural machine translation. *Transactions of the Association for Computational Linguistics*, 6:635–649.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017a. Word sense disambiguation: A unified evaluation framework and empirical comparison. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 99–110, Valencia, Spain. Association for Computational Linguistics.

Alessandro Raganato, Claudio Delli Bovi, and Roberto Navigli. 2017b. Neural sequence learning models for word sense disambiguation. In *Proceedings of the 2017 Conference on Empirical Methods*

in Natural Language Processing, pages 1156–1167, Copenhagen, Denmark. Association for Computational Linguistics.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Bianca Scarlini, Tommaso Pasini, and Roberto Navigli. 2020. With more contexts comes better performance: Contextualized sense embeddings for all-round word sense disambiguation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3528–3539, Online. Association for Computational Linguistics.

Benjamin Snyder and Martha Palmer. 2004. The English all-words task. In *Proceedings of SENSEVAL-3, the Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43, Barcelona, Spain. Association for Computational Linguistics.

Loïc Vial, Benjamin Lecouteux, and Didier Schwab. 2019. Sense vocabulary compression through the semantic knowledge of WordNet for neural word sense disambiguation. In *Proceedings of the 10th Global Wordnet Conference*, pages 108–117, Wroclaw, Poland. Global Wordnet Association.

Warren Weaver. 1949. Translation. In *Machine Translation of Languages: Fourteen Essays (written in 1949, published in 1955)*, pages 15–23, New York, NY. W. N. Locke and A. D. Booth, Eds. Technology Press of MIT, Cambridge, MA, and John Wiley & Sons.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLNet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced language representation with informative entities. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1441–1451, Florence, Italy. Association for Computational Linguistics.

A Transformer Architectures

In this Section we show the results attained when using RoBERTa$_{large}$ (Liu et al., 2019) and XLNet$_{large}$ (Yang et al., 2019) as pretrained model for ESCHER. We note that 73 training examples out of 226,036 could not fit in 512 bpes, i.e., the maximum input length for both models, and we therefore discard them. In Table 6, we report the results on the English WSD evaluation framework of Raganato et al. (2017a) for ESCHER initialized with the aforementioned models along with its performance when using BART$_{large}$.

ESCHER Transformer	Model Parameters	SE07	ALL
RoBERTa$_{large}$	355M	76.0	80.5
XLNet$_{large}$	340M	76.2	80.6
BART$_{large}$	406M	76.3	80.7

Table 6: ESCHER results on ALL when initialized with different transformers.

B Training Details

We use BART$_{large}$ as transformer architecture which consists of 12 encoder layers and 12 decoder layers with 1024 hidden size. We train the model with a constant learning rate of 0.00001, Rectified Adam as optimizer and a batch size of 700 tokens. We accumulate the gradient for 20 steps and clip it at 10. The model is trained for a maximum of $300,000$ steps. We compute the F1 score on the validation dataset every 2000 steps and stop the training if the model does not improve for 15 consecutive tests ($30,000$ steps). The whole training is done with half precision and an amp-level of O1. It is worth noting that the training of ESCHER on SemCor (Miller et al., 1990) took less than 5 hours on a GeForce RTX 2080ti.