Unsupervised Mapping of Arguments of Deverbal Nouns to Their Corresponding Verbal Labels

Aviv Weinstein
Department of Computer Science
Bar-Ilan University
aviv.wn@gmail.com

Yoav Goldberg
Department of Computer Science
Bar-Ilan University
yoav.goldberg@gmail.com

Abstract

Deverbal nouns are nominal forms of verbs commonly used in written English texts to describe events or actions, as well as their arguments. However, many NLP systems, and in particular pattern-based ones, neglect to handle such nominalized constructions. The solutions that do exist for handling arguments of nominalized constructions are based on semantic annotation and require semantic ontologies, making their applications restricted to a small set of nouns. We propose to adopt instead a more syntactic approach, which maps the arguments of deverbal nouns to the universal-dependency relations of the corresponding verbal construction. We present an unsupervised mechanism—based on contextualized word representations—which allows to enrich universal-dependency trees with dependency arcs denoting arguments of deverbal nouns, using the same labels as the corresponding verbal cases. By sharing the same label set as in the verbal case, patterns that were developed for verbs can be applied without modification but with high accuracy also to the nominal constructions.

1 Introduction

Systems that aim to extract and summarize information from large text collections often revolve around the concept of predicates and their arguments. Such predicates are often realized as verbs (*the performers interpret the music*), but the same predicative concepts can also be realized as nouns (*musical interpretation by the performers*). This process of realizing verbal predicates as nouns is called *nominalization*, and it involves changing the syntactic structures around the content words participating in the construction, while keeping its semantics the same. In this work, we are interested in mapping arguments of nominal constructions that appear in text, to the corresponding ones in verbal structures (i.e., to identify the syntactic object role of *music* and syntactic subject role of *performers*, in *music interpretation by the performers*).

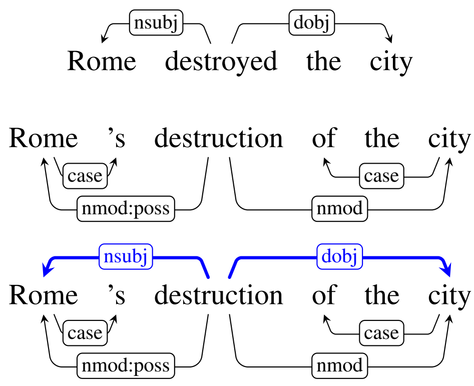


Figure 1: Example of our task. Top: verbal argument structure. Middle: nominal argument structure. Bottom: nominal structure enriched with corresponding verbal argument labels (thick blue edges).

Nominalizations, also known as nominal predicates, are nouns derived from words of a different part of speech, such as verbs or adjectives. For example, in English[1], the nominalization *interpretation* is derived from the verb *interpret*, and the nominalization *precision* is related to the adjective *precise*. The usage of nominalizations is widespread in English text, and according to Gurevich et al. (2007), about half of all sentences in written texts contain at least one nominalization. In our work, we observed a ratio of 120k nominalizations to 180k verbs, in a random collection of 100k Wikipedia sentences. Thus, interpretation of nominalizations is central to many language understanding tasks. In the current work, We focus on nominalizations which are derived solely from verbs, commonly called deverbal nouns.

Existing attempts around identifying arguments of nominalizations either rely on a predefined semantic roles ontology (e.g., SRL based roles such

[1]While this work focuses on English nominalizations, the phenomena itself is not English specific.

Findings of the Association for Computational Linguistics: ACL 2023, pages 2921–2935

as those in VerbNet (Schuler, 2005) or FrameNet (Baker et al., 1998)) as suggested by Pradhan et al. (2004), Padó et al. (2008) and Zhao and Titov (2020), or consider a limited subset of nominalized structures (Lapata (2000) and Gurevich and Waterman (2009)). Early works approached the task in a fully supervised manner (Lapata (2000), Pradhan et al. (2004)), hence suffering from insufficient annotated nominal data. To overcome that, Padó et al. (2008) and more recently Zhao and Titov (2020) considered a transfer scenario from verbal arguments to nominal arguments while assuming only supervised data for verbs. Nevertheless, their methods were limited to specific predicates, even with extensive annotated verbal data. Moreover, the previous works considered each a different set of argument types due to supervision constraints.

Our Proposed Task Rather than relying on a predefined semantic roles ontology, in this work we propose to map the arguments of deverbal nouns to the *syntactic* arguments of the corresponding active verbal form. This allows us to define a task with a consistent and a restricted label set (syntactic subject, syntactic object, syntactic prepositional modifier with preposition X), while still maintaining expressivity: if one knows how to extract the verbal argument from the active verbal form, they will be able to also extract the nominal ones.

A natural formulation is to ask "How will this verb arguments be realized in a deverbal noun construction?". However, this approach is problematic, as the same verbal structure, e.g. *IBM appointed Sam as manager*, can be realized in many different ways around the same nominalization, including: *IBM's appointment of Sam as manager*, *Sam's appointment as manger by IBM* and *Sam's IBM appointment as manager*.

One solution would be to ask for all the possible nominal realizations. This is the approach taken by nominalization lexicons such as NomLex (Macleod et al., 1998). However, this is also problematic in practice, as the different possible syntactic structures may conflict when encountering a nominalization within a sentence (*IBM's appointment* vs. *Sam's appointment*).

We resolve this by asking the opposite question: "given a nominalized instance within a sentence and its set of arguments, how will these arguments map to those of an active verb construction?". That is, rather than asking "how will this verbal construction be realized as a nominal one" we ask "how will this nominal case be realized as an active verb construction". Using this formulation, we define a corpus enrichment task, in which we take in a corpus of syntactic trees, and annotate each deverbal noun case with its nominal arguments, using the corresponding verbal argument labels. An example of the trees enrichment is provided in Figure 1.

Potential Utility Our motivation follows that of Tiktinsky et al. (2020): we imagine the use of the enhanced trees in systems that integrates universal dependency trees (Nivre et al., 2016) as part of their logic, using machine-learned or pattern-based techniques. Our proposed enrichment will allow users to search for a verb construction, and retrieve also nominal realizations of the same relation.

One proposed usage case regards the task of Open Information Extraction (OpenIE; Etzioni et al., 2008), which refers to the extraction of relation tuples from plain text, without demanding a predefined schema. These tuples can be extracted from both verbal and nominal phrases, e.g., the tuple (Steve Jobs; founded; Apple) from the phrase *Steve Jobs founded Apple* and the tuple (IBM; research) from the phrase *IBM's research*. Some OpenIE systems, such as Renoun (Yahya et al., 2014) and Angeli et al.'s (2015) system, integrate rule-based patterns to extract such relations from nominal phrases, e.g., (X; Y) from phrases of the structure "X's Y". However, these patterns can be misleading, as *IBM's research* interprets differently from *Rome's destruction* (IBM researched vs. Rome was destructed), leading to contradicting relations. To overcome that, we suggest using verb-based patterns to extract relations from nominal phrases, upon integrating our enhanced trees. Concretely, based our enhanced trees, an OpenIE system can use a pattern that detects the nsubj-phrase and dobj-phrase for both verbs and nouns, to construct the relation tuple (nsubj; verb/noun; dobj). With this approach, different nominal phrases with the same syntactic structure, would properly map to different ordered relations, as (destruction; Rome) for the phrase *Rome's destruction*.

An Unsupervised Approach We take an unsupervised approach to this nominal-to-verbal argument mapping, relying on pre-trained contextualized word representations. The intuition behind our approach is that in order to resolve nominal arguments to verbal ones, there are two prominent signals: the semantic types of the arguments, and their syntactic configuration with respect to their predicate. We hypothesize that pre-trained contextualized word embeddings capture both of these signals (as shown in Section 7.2), and also capture the similarities between the verbal and nominal cases (as demonstrated in Appendix A). Briefly, our approach works by identifying the candidate arguments of each deverbal noun instance, retrieving a set of sentences containing the corresponding active verb form, encoding both the deverbal noun instance and the active verb sentences using a masked language model, and searching for a mapping that maximizes some similarity metric between the nominal argument candidates and the verbal instances.

Our contributions in this work are thus twofold: (1) we formulate the task of aligning nominal arguments to the arguments of their corresponding active verbal form; and (2) we propose an unsupervised method for tackling this task. We also provide code[2] for enriching universal dependency trees (Nivre et al., 2016) with nominal arguments.

2 Deverbal Nouns

Deverbal nouns are one type of nominalizations which are derived specifically from verbs, e.g., the deverbal noun *treatment* is derived from the verb *treat*. The events represented by deverbal nouns are described using phrases in the sentence that complement the nouns. The arguments of the deverbal noun correspond to the arguments of the matching verb; each matches a different question about the action taken. For instance, in the phrase *professional treatment of illness*, *professional* refers to the actor/subject of the verb *treat* (professionals), and *illness* refers to the object of the action *treat*.

The deverbal nouns, as typical nouns, are most often complemented by other noun phrases (*treatment of illness*, *his treatment* and *health treatment*) and adjectives (*professional treatment*). Implicit and other types of complementing arguments are not considered part of this work's scope. Each deverbal noun defines a unique structure of these arguments, assigning different roles for the same typed arguments. For instance, consider the phrases *time preference of the individual* and *individual waste of time*, which match the same syntactic structure ("noun-compound *of* noun"). However, the first sentence matches the structure "Obj Noun of Subj" ("individuals$_2$ prefer time$_1$"), and the second sentence refers to the structure "Subj Noun of Obj" ("individual$_1$ waste time$_2$"). Furthermore, even the same deverbal noun may demand different labels for similar arguments in different contexts. For example, in the phrase "*Rome's destruction*", *Rome* was destroyed, whereas in the phrase "*Rome's destruction of the city*", *Rome* is the destroyer. Therefore, the argument roles are not determined solely by syntactic structure, and incorporate a mix of syntactic configuration, argument semantics, and predicate-specific information.

3 Related Works

Arguments of nominalizations were long investigated in the field of NLP. One early research explored the syntactic structure of the arguments and modeled the structure of many nominalizations, resulting in a detailed lexicon called NomLex (Macleod et al., 1998). The lexicon seeks to describe the allowed complements structures for a nominalization and relate the nominal complements to the arguments of the corresponding verb. Following the publishing of NomLex, Meyers et al. (1998) described how an Information Extraction (IE) system could exploit the linguistic information in the NomLex lexicon. Yet, the suggested approach remained hardly utilized by further research, as many works only exploited the verb-noun pairs specified by the lexicon.

Regarding identifying and labeling nominalization's arguments, a supervised approach was suggested while considering various task settings. One preceding paper by Lapata (2000) presented a probabilistic procedure to infer whether the modifier of a nominalization (the head noun) stands in subject or object relation with it. For instance, the algorithm should predict that the modifier's role in the phrase *child behavior* is subject since the phrase refers to the *child* as the agent of the action described by the verb *behave*. Stated differently, this procedure focuses on extracting only one specific argument of nominalizations in a noun phrase. Another distinguished paper by Pradhan et al. (2004) considered FrameNet-based (Baker et al., 1998) semantic arguments of nominalizations and applied a machine learning framework for eventive nominalizations in English and Chinese, aiming to identify and label their arguments. Finally, Kilicoglu et al. (2010) published a similar approach for nominalizations used in biomedical text.

[2]Our code is available at https://github.com/AvivWn/NounVerbUDTransfer

Some related works acknowledge the shortage of labeled argument nominalizations and suggest unsupervised methods for data expansion based on labeled argument verbs. Similarly to ours, these works exploited the similarity and alignment of the noun-verb arguments. For example, Padó et al. (2008) and Zhao and Titov (2020) considered the argument labeling task for nominalizations in a setup where the verbal sentences are human labeled, and with regards to semantic role labeling (SRL) arguments. Padó et al. (2008) exploited the similarities between the argument structure of event nominalizations and corresponding verbs while utilizing common syntactic features and distributional-semantic similarities. More recently, Zhao and Titov (2020) suggested a variational auto-encoder method, in which the labeler serves as an encoder, whereas the decoder generates the selectional preferences of the arguments for the predicted roles.

A different approach taken by Gurevich and Waterman (2009) using a fully unsupervised manner while automatically extracting and labeling verbal arguments of verbs from a large parsed corpus of Wikipedia. This approach resembles an intermediate stage of ours yet differs as it considers a reduced set of argument types (subject and object) and a reduced possible set of argument syntax for the nominalizations (possessive and 'of' arguments). Lately, Lee et al. (2021) engaged with a different task with similar applications. They suggested an unsupervised method for paraphrasing clauses with nominalizations into active verbal clauses.

4 Task Definition

As discussed in the introduction, we define a task of labeling the arguments of deverbal nouns within a sentence, with labels of the arguments in the corresponding active verb constructions. Here we provide a more complete and formal definition. While our aim is to label all of the deverbal nouns in a given corpus, here we focus on describing the task with relation to a single instance of a sentence and a deverbal noun within it.

We consider the syntactic arguments of active verbal forms to belong to the set L consisting of the universal dependency relations *nsubj*, *dobj* and *nmod:X*, where X is a preposition (e.g., *nmod:in*, *nmod:on*, *nmod:with*). In words, the syntactic subject, syntactic object, and arguments attached as prepositional phrases where the identity of the preposition is part of the relation. While these prepositions may correspond to many different semantic roles, for a given verb they usually indicate a concrete and unique role.

Formally, given a sentence with words $w_1, \dots, w_n$, and a marked deverbal noun within the sentence (say in position w_i), we seek to find K pairs of the form (rel_k, w_{j_k}), $1 \leq k \leq K$, where $rel_k \in \{nsubj, dobj, nmod{:}X\}$ and w_{j_k} is a word in the sentence (j_k is an index of a sentence word). For simplicity, we also demand that every relation type cannot be repeated more than once in the identified set of pairs. These pairs indicate arguments of the deverbal noun and their relations to it, expressed using an active-verb label set.

In Figure 1, the blue edges of the bottom tree indicate the output *(nsubj, 1), (dobj, 6)*. Note that the task includes both the *identification* of the arguments and their *label assignment*.

5 Methodology

While we intend to handle all deverbal nouns in a given collection of sentences, here we focus on how to resolve a single deverbal noun. We identify deverbal nouns and their corresponding verbal forms based on a given lexicon of verb-noun pairs, which we consider as input. In this work, we use the NomLex lexicon (Macleod et al., 1998), where future work can also replace this with a learned model.

Given a deverbal noun within a sentence, we first identify its potential arguments. This is realized by searching a set of syntactic relations in the corresponding universal dependency tree (we use the UDv1 parser trained by Tiktinsky et al. (2020) via the spaCy toolkit[3]). We then label the arguments by comparing their contextualized word embeddings to those of the corresponding verb arguments, in a set of sentences containing this verb (we further motivate this comparison in Appendix A). Finally, based upon the labeled arguments, we construct the final output as pairs of the arguments' label (i.e. verbal UD relation) and the arguments' head word.

5.1 Argument Identification

Given a sentence and a specific deverbal noun within, we first identify the phrases which could correspond to the desired arguments of the matching verb. The identified set of phrases is referred to as "argument candidates". Naively, every phrase in the sentence can complement the deverbal noun

[3] https://spacy.io

$$\ell_n = \arg\max_{\ell} sim(\mathbf{a_n}, avg(\{\tilde{\mathbf{a}} \mid \ell(\tilde{a}) = \ell, \tilde{a} \in \tilde{A}\})) \tag{1a}$$

$$\ell_n = \arg\max_{\ell} sum(\{sim(\mathbf{a_n}, \tilde{\mathbf{a}}) \mid \ell(\tilde{a}) = \ell, \tilde{\mathbf{a}} \in knn(\mathbf{a_n}, \tilde{\mathbf{A}}, k)\}) \tag{1b}$$

and be considered as an argument, thus resulting in a relatively large set of candidates. To reduce this set, we consider the syntactic dependency tree of the sentence, searching for words that stand with direct dependency relation with the deverbal noun. Then, for every identified word we construct the argument candidate as the phrase corresponding to the subtree headed by this word according to the dependency tree. More specifically, we observed that arguments of deverbal nouns are realized using words that stand with the deverbal nouns in a small set of possible syntactic relations: *nmod:poss*, *compound*, *amod*, and *nmod:X*. Table 1 provides an example of these syntactic relations, using argument candidates for the deverbal noun *analysis*. In Section 7.1 we compare this approach and other considered approaches to identify the arguments.

Phrase	UD Relation
his *analysis*	nmod:poss
data *analysis*	compound
linguistic *analysis*	amod
analysis of ***the data***	nmod:of

Table 1: The types of UD relations we used to identify candidate arguments, and their example with the deverbal noun *analysis*.

5.2 Argument Labeling

Upon argument identification, we aim to label the identified argument candidates of the deverbal nouns, with the desired argument types (*nsubj*, *dobj*, *nmod:X* or $\emptyset$), such that the labels align to the labels of the corresponding arguments in the active verbal form (the label $\emptyset$ indicates that this argument candidate is not in fact an argument of the noun, such as *primary* in the phrase *the primary influence*). For instance, in the sentence *The emperor's destruction of Paris*, we wish to label *the emperor* as *nsubj* and *Paris* as *dobj*, since the sentence can only be understood as the verbal sentence *The emperor destroyed Paris*.

Concretely, denote the argument candidates as $a_1, \ldots, a_N$. We need to assign them with labels $\ell_1, \ldots, \ell_N$, where $\ell_i \in \{\emptyset, nsubj, dobj, nmod{:}X\}$, under the constraint that every two arguments a_i, a_j, can share labels if and only if they match the label $\emptyset$ (as emphasized in the defined task).

We start from obtaining a set of verbal reference sentences S, containing M sentences $s_1, \ldots, s_M$, each sentence s_m contains the verbal form of the deverbal noun (these are obtained using a simple keyword search). In each of these instances s_m, we use simple active and passive verbal dependency patterns to identify the A_m verbal arguments $\tilde{a}^m_1, ..., \tilde{a}^m_{A_m}$, labelled as $\tilde{\ell}^m_1, \ldots, \tilde{\ell}^m_{A_M}$. Intuitively, we now seek to find for each of our nominal argument a_n the most similar verbal argument $\tilde{a}^m_j$, and match their labels. In our experiments, we obtained a set S containing about 1,500 reference sentences[4] regarding every verb that was required by the evaluation datasets.

We encode both the input sentence and the reference sentences using a contextualized encoder (we use BERT-large-uncased (Devlin et al., 2018) in this work), resulting in vectors $\mathbf{a_1}, \ldots, \mathbf{a_N}$ for the input sentence and vectors $\tilde{\mathbf{a}}^{\mathbf{m}}_{\mathbf{1}}, ..., \tilde{\mathbf{a}}^{\mathbf{m}}_{\mathbf{A_m}}$ for each verb reference sentence s_m. We denote the entire set of verbal arguments as $\tilde{A}$ and the corresponding set of vectors as $\tilde{\mathbf{A}}$. We use a metric function $sim(\mathbf{a}, \tilde{\mathbf{a}})$ over the pair of vectors to quantify their similarity (we use *cosine* similarity in this work). We then choose the label of each nominal argument a_n independently[5] based on its closest neighbours in $\tilde{\mathbf{A}}$. We consider two variants: in the first one (1a, nearest-avg-argument), we select the label ℓ_n by averaging the reference vectors for each verbal argument label, and then choosing the label whose corresponding average vector is the most similar to the nominal argument's vector. In the second variant (1b, k-nearest-argument), we take the k-nearest verbal argument vectors (we use k=5) to the nominal argument vector. We compute the sum of similarities between $\mathbf{a_n}$ and each of the k-nearest vector $\tilde{\mathbf{a}}$ corresponding to each label, and choose

[4] We considered $\ll$1,500 reference sentences for less frequent verbs.

[5] We also experimented with jointly labeling several arguments, but did not see any benefit.

the label with the highest sum.

For both labeling variants, we assign the label $\emptyset$ for arguments whose similarity with any other reference argument does not pass a chosen threshold.

6 Evaluation Data

Our task is to identify arguments of deverbal nouns and assign each one of them a label from the set $L = \{nsubj, dobj, nmod{:}X\}$. For evaluation, we need sentences with deverbal nouns whose arguments are labeled with these relations. For example, the deverbal noun *relocation* in the phrase *Family relocation to Manchester* should be labeled with the pairs *(nsubj, 1)* and *(nmod:to, 4)*, as specified in Section 4.

We create three such evaluation datasets, the first based on a nominalization paraphrasing dataset, and the other two are based on the NomLex lexicon, while they differ by the coverage of deverbal nouns that they consider, as we further explain. Moreover, to compare our method's performance to earlier works, we consider the CoNLL-2009 dataset (Hajič et al., 2009) for evaluation, as we discuss in 7.3.

The paraphrasing-derived evaluation set is derived from a manually annotated dataset for the task of paraphrasing sentences from nominal to verbal form (Lee et al., 2021). The original dataset includes a collection of 449 samples from 369 unique sentences representing 142 different verbs. Each sample represents a paraphrasing between the original nominalization phrase (from a given sentence) and a verbal clausal phrase, for instance *genetic analysis from a sample* which is paraphrased as *analyze genes from a sample*. For every paraphrasing sample, the dataset specifies the components of the nominal phrase within the structure "*adj/noun* nominalization *prep pobj*", and the components of the active verbal phrase ("*arg0* verb *arg1 pp*").

To construct our evaluation set based on this data, we first match each of the nominal components adj/noun and pobj with a verbal component from the set of arg0, arg1 and pp, choosing the one with the closest orthography to the nominal one. From this, we derive the verbal argument labeling for the components of the nominal phrase. Then, we replace each verbal label with its matching UD relation.[6] Finally, for every nominal component we determine its head word position in the given context. The word positions paired with the matching verbal relations, construct a sample in our new paraphrasing-derived evaluation set.

In the course of dataset construction, we filter out some data samples. To start with, data samples that specify two nominal components that match the same verbal component were removed from our dataset, as they do not fit the constraints of the defined task. For example, in the phrase *environmental assessment for the project* the combined components of the noun can be understood together as the object of the matching verb (*assess the environmental impact of the project*), hence resulting with two nominal arguments labeled with the same verbal relation. Secondly, we consider only the first single data sample for every repeated nominal phrase to ensure a single truth of labeling for every nominal phrase. Following the filtering process we remain with 309 samples with 122 different verbs.

The NomLex evaluation sets are constructed using the NomLex lexicon.[7] The NomLex lexicon contains a list of about 4k deverbal nouns, and for each of them specifies the various ways in which their arguments can be realized syntactically, and how they map to the corresponding verbal arguments. For example, an adapted NomLex entry for a deverbal noun like *destruction* would specify the related forms of the noun (i.e., the verb and other related deverbal nouns) and, most significantly, a set of dependency-tree patterns corresponding to several different realizations of the noun. Each dependency-tree pattern represents a set of labeled arguments in a specific dependency tree. For instance, the entry of *destruction* would contain a pattern that corresponds to the dependency structure shown in the middle of Figure 1 and demands the labeling of *Rome* as subject and *city* as object. Hence, using a parsed dependency tree of a sentence with a deverbal noun, we can extract the labeled arguments in the sentence for any specified pattern that fulfills the sentence's dependency structure. However, this method does not allow for a definitive decision in many cases, as the lexicon often contains multiple labeled contradicting patterns. In Section 7 we show that relying solely on NomLex results in a significantly lower precision.

We collect English Wikipedia sentences from

[6] *arg0* $\mapsto$ *nsubj*, *arg1* $\mapsto$ *dobj*, *pp* $\mapsto$ *nmod:X*, where X is determined by the leading preposition.

[7] We converted the NomLex lexicon from its original LISP-based formatting and phrase-structure trees, to a more modern form encoded in JSON and using UD syntactic relations. The code for this conversion is accessible at `https://github.com/AvivWn/NounVerbUDTransfer`.

Guo et al. (2020) that contain a deverbal noun, and for each sentence, we identify the deverbal noun's arguments and labels based on the adapted NomLex entry as described above. We discard sentences for which the entry suggests two or more different assignments, when matching two or more dependency patterns. We then map NomLex's labels into the corresponding dependency relations of the active verbal form. To match the examples in the paraphrasing dataset, we consider only data samples with two labeled arguments each. We divide the collected samples into two evaluation sets based on the verbal form of the represented deverbal nouns. **NomLex**$_{paraphrasing}$ considers only samples which refer to verbs that appeared in the paraphrasing-derived corpus, whereas **NomLex**$_{other}$ considers samples that match 315 other verbs. In each evaluation set, we keep 25 labeled sentences for each verb.

Tune/Test Split Our method is unsupervised but still requires tuning of hyperparameters. We keep a tuning subset for each origin of the evaluation set (paraphrasing-derived and NomLex), which is also used for evaluation during development. In the paraphrasing dataset, we sample 20% of the dataset to construct the tuning set while keeping aside 80% of the dataset for evaluation. Out of the 122 verbs in the paraphrasing-derived evaluation set, 12 appear only in the tuning set, 83 only in the test set, and 27 appear in both sets. The split aims to ensure that the results are not verb-specific and to prevent overfitting, as we do hyperparameter optimization on the tuning set, which does not contain all the verbs that appear in the test set. To tune the method for NomLex-based data, we perform a similar tune-test split on NomLex$_{paraphrasing}$ based upon the same tune-test verb division made for the paraphrasing evaluation set. Concretely, NomLex instances of the 12 tuning-only verbs and 83 test-only verbs were included only in the NomLex tuning set and test set, correspondingly; Instances of the 27 common verbs were divided into the tune-test sets in a 20%-80% ratio. Moreover, we preserve entirely NomLex$_{other}$ corpus for testing.

Evaluation Metrics We use two evaluation metrics: **Relation-F1** is the F1 score of all the predicted word-relation pairs compared to the gold labeled pairs (without distinguishing argument labels, for comparability with Zhao and Titov (2020) which uses CoNLL-2009 evaluation scorer (Hajič et al., 2009)). **Exact-Match** scores how many noun instances had all their relations identified and labeled correctly. A predicted relation is considered correct if it matches both the same argument head word and the same label as the gold relation.

7 Experiments and Results

In this section, we consider the results of our method on the evaluation sets and experiments we conducted concerning the two stages of our method. The setup which produced the best results is discussed in 7.2, including the chosen hyperparameters, which were tuned over the tuning sets.

Baseline As a baseline for our approach, we considered the same process we used for generating the NomLex evaluation sets. More specifically, for a given parsed sentence with a given deverbal noun, our baseline method attempts to match the deverbal noun instance with all dependency patterns in appropriate entry within the adapted NomLex lexicon. Every fulfilled pattern should result in a set of labeled arguments. The combined set of non-colliding arguments, i.e., arguments that match a single argument type, are then mapped into pairs of headwords and UD relations, which are also the output of the baseline method.

7.1 Argument Identification

Using the set of relation labels in Section 5.2 and considering each one of them as an argument candidate, we cover 94.6% of all the relations in our paraphrasing-derived test-set, while producing 76 candidates (16.2% of all proposed candidates) that are not arguments. We find this to be of sufficient coverage and accuracy for the paraphrasing dataset. Regarding the NomLex evaluation sets, all arguments were identified using that relations set (100% coverage), while producing 24.8% and 23.1% non-argument candidates for NomLex$_{paraphrasing}$ and NomLex$_{other}$, respectively. As NomLex does not consider adjectival arguments, we choose to consider a reduced set of dependency relations without the *amod* relation, keeping the same coverage and producing only 8.8% and 8.7% non-argument candidates, respectively.

For the paraphrasing-derived dataset we also considered two other alternatives: relying on the information in the NomLex lexicon for each noun, resulting in coverage of 58.5% and producing 6.9% non-argument candidates, and relying on NomLex

Method	Paraphrasing-derived F1	Paraphrasing-derived Exact	NomLex$_{paraphrasing}$ F1	NomLex$_{paraphrasing}$ Exact	NomLex$_{other}$ F1	NomLex$_{other}$ Exact
baseline (NomLex-based)	43.42	7.66	-	-	-	-
all-subject	27.67	0.00	37.04	0.00	41.52	0.00
all-object	36.50	0.00	40.24	0.00	38.19	0.00
nearest-avg-argument	44.08	17.74	39.81	18.38	40.10	19.49
k-nearest-argument	**62.93**	**36.29**	**53.74**	**34.98**	**53.67**	**35.06**

Table 2: The best results of the two suggested labelers on the three test sets, compared to the baseline process and the naive methods. Regarding metrics, 'F1' refers to Relation-F1 and 'Exact' refers to Exact-Match.

lexicon while also considering *amod* relations, resulting in an increased coverage (85.3%) and increased non-argument candidates (13.9%). These low coverage results are anticipated as NomLex lexicon lacks the representation of some nominal structures, hence we chose the label-set approach as it was the most effective one.

We explored the resulted argument candidates and gathered three main reasons for the non-argument candidates. First, some correspond to arguments missing in the evaluation set. In the paraphrasing set, this is due to the focus on two arguments structure for each deverbal noun; In contrast, in the NomLex evaluation sets, this is primarily due to discarding of undetermined arguments and for the lack of prepositional adjuncts representation (which are captured using the dependency relations). Other resulted non-argument candidates are misaligned with the correct arguments, not sharing the same head-word, as emerged from a human-based evaluation set (such as paraphrasing-derived). Finally, the remaining non-arguments are indeed not an argument of the noun.

7.2 Argument Labeling

Main Results We experiment with two different labeling methods, as discussed in Section 5.2: nearest average of reference argument representations for each argument (nearest-avg-argument); k-nearest reference arguments (k-nearest-argument). The results of the various labeling methods are shown in Table 2 while considering the most suitable identification method for every evaluation set as raised from the argument identification comparison. We report our results on the three test sets and in comparison with the performance of the baseline method and naive 'all-subject' and 'all-object' methods (which label all argument relations with *nsubj* and *dobj*, respectively). As articulated from our results, both labeling methods performed better than the baseline regarding the paraphrasing evaluation set. Moreover, k-nearest-argument outperformed nearest-avg-argument on all metrics of all evaluation sets. Best results were attained by calibrating the methods on the matching tuning sets, e.g., selecting a specific threshold for labeling $\emptyset$-typed arguments (0.56 for paraphrasing tune-set and 0.48 for NomLex tune-set). Yet, we examined similar performance tendencies between the tuning sets and the test sets (see Appendix B), implying a generalization of our method for other examples. We further validated our method generalization for any arbitrary verb, by scoring relatively similar results on NomLex$_{other}$ as on NomLex$_{paraphrasing}$ without additional tuning, while each considers nouns that match a different set of verbs. The extended results in Appendix B also demonstrate the Relation-F1 scores of our best method regarding the most common relations in the test sets.

Importance of Contextualization Arguments of verbs and deverbal nouns share semantics, as both commonly paraphrase the same entity in different contexts. For instance, the subject of the verb *acquire* usually matches the semantic role of a 'HUMAN' (*John acquired the ingredients*) or a 'COMPANY' (*Apple acquired another startup company*). The same subjects can be realized in a deverbal noun context, as in *The ingredients acquisition of john* and *Apple's acquisition of the startup company*, correspondingly. The semantic role of words can be represented by vector representations, both contextualized representations as BERT and uncontextualized representations as Word2Vec (Mikolov et al., 2013) vectors. We compared our main results with pre-trained BERT-based representations to uncontextualized representations, using pre-trained Fasttext Word2Vec model made by Bojanowski et al. (2017). The results of our method regarding the two representations are shown in Table 3. Using Word2Vec we see a decrease of about 25% in Relation-F1 and about 40% in Exact-Match compared to BERT results using our best method, from which we conclude that the context of the argument also affects the performance of our method.

Method	BERT	Word2Vec
nearest-avg-arg	**44.08 (17.74)**	20.78 (4.44)
k-nearest-arg	**62.93 (36.29)**	46.53 (21.37)

Table 3: The best results of the suggested labelers using BERT and Word2Vec representations, on the paraphrasing test set, specified as "Relation-F1 (Exact-Match)".

Syntax vs Semantics The previous experiment has demonstrated that the contextualized vectors outperform the static ones, suggesting the need for more than word semantics. In the following experiment, we further quantify the contribution of syntactic position vs. argument semantics to the final predictions. We manipulate the paraphrasing evaluation set by switching the sentence positions of the two specified arguments for each tagging sample. Note that the resulting sentence is usually neither grammatically nor semantically correct. Then, we apply our labeling stage while considering the BERT vectors over the arguments in the new positions. When compared to the labels of the same arguments received in the original positions, we see almost 70% difference. Thus, the syntactic position has an innegligible effect on the verb-noun alignment that our method aims to resolve.

7.3 Comparison to Earlier Work

Existing unsupervised attempts that approach the nominal argument labeling task as a transfer scenario from verbal arguments to nominal arguments (as our work), rely on a predefined semantic roles ontology. For instance, Zhao and Titov (2020) consider SRL roles of verbs to label nouns with the same set of roles, as appears in CoNLL-2009 dataset (Hajič et al., 2009). Our defined task and proposed methods do not require a predefined semantic roles ontology, yet can be tested on one for comparability with such existing work. Thus, we apply our labeling methods on CoNLL-2009 nominal test data after verbalizing the nominal predicates in the dataset while considering the CoNLL-2009 verbal train data as verbal references. For evaluation comparability with Zhao and Titov (2020), we skip the argument identification stage and assume the identified arguments are given. Finally, we calculate the F1 performance (as discussed for "Relation-F1" in Section 6) of our methods, which we compare to the matching ones reported by Zhao and Titov (2020). As shown in Table 4, our best method ('k-nearest-argument') outperforms their baselines ('Most-frequent', 'Factorization' and 'Direct-transfer'). However, their 'Full-system' approach transcends our method by exploiting a supervised verbal SRL system and data augmentations, which we do not use in our work.

Method	F1
Most-frequent	56.51
Factorization	44.48
Direct-transfer	55.85
Full-system	**63.09**
k-nearest-argument (Ours)	58.82

Table 4: F1 results reported by Zhao and Titov (2020) on CoNLL-2009 nominal test data, compared to the result of our best labeler applied on the same dataset.

8 Conclusions

In this work, we formulate the task of aligning arguments of deverbal nouns to the arguments of their corresponding active verbal form. We formulate the task as a UD enrichment task, aiming to enrich deverbal nouns in text with verbal UD relations for the matching nominal arguments. Our formulation, compared to the ones suggested in previous works, does not rely on a predefined roles ontology.

We suggest an unsupervised approach to this nominal-to-verbal argument mapping based on pretrained contextualized word representations. Our method tries to match nominal identified arguments with automatically extracted arguments of the corresponding verb. The suggested method outperforms the NomLex-based baseline, which is based on an expertly constructed comprehensive lexicon. We also show the importance of contextualization, experiencing a 25% decrease in performance when using uncontextualized vectors. Moreover, we further validate our hypothesis that semantics and syntactic structure are captured in the considered word representations using a dedicated experiment.

We provide a standalone code for enriching universal dependency trees with nominal arguments for a given parsed corpus, which can be integrated into NLP systems that use universal dependency patterns as part of their design or features.

Limitations

The main drawback of the work is in its evaluation, which was performed on datasets which were not manually annotated for the task, but adapted to it in various means. While we believe these evaluation sets do provide a strong indication regarding task performance, evaluating on bespoke data explicitly annotated for the task is usually preferable. Another limitation is language specificity: the work currently focuses on English, without considering other languages, which are also left for future work.

Ethics Statement

Like all works that depend on embeddings, the resulting models may be biased in various ways. Users should take this into consideration when deploying them in products.

Acknowledgements

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme, grant agreement No. 802774 (iEXTRACT).

References

Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D Manning. 2015. Leveraging linguistic structure for open domain information extraction. 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)*, pages 344–354.

Collin F Baker, Charles J Fillmore, and John B Lowe. 1998. The berkeley framenet project. In *36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1*, pages 86–90.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Oren Etzioni, Michele Banko, Stephen Soderland, and Daniel S Weld. 2008. Open information extraction from the web. *Communications of the ACM*, 51(12):68–74.

Mandy Guo, Zihang Dai, Denny Vrandečić, and Rami Al-Rfou. 2020. Wiki-40b: Multilingual language model dataset. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 2440–2452.

Olga Gurevich, Richard Crouch, Tracy Holloway King, and Valeria De Paiva. 2007. Deverbal nouns in knowledge representation. *Journal of Logic and Computation*, 18(3):385–404.

Olga Gurevich and Scott Waterman. 2009. Mining of parsed data to derive deverbal argument structure. In *Proceedings of the 2009 Workshop on Grammar Engineering Across Frameworks (GEAF 2009)*, pages 19–27, Suntec, Singapore. Association for Computational Linguistics.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štepánek, et al. 2009. The conll-2009 shared task: Syntactic and semantic dependencies in multiple languages.

Halil Kilicoglu, Marcelo Fiszman, Graciela Rosemblat, Sean Marimpietri, and Thomas C Rindflesch. 2010. Arguments of nominals in semantic interpretation of biomedical text. In *Proceedings of the 2010 workshop on biomedical natural language processing*, pages 46–54. Association for Computational Linguistics.

Maria Lapata. 2000. The automatic interpretation of nominalizations. In *AAAI/IAAI*, pages 716–721.

John Lee, Ho Hung Lim, and Carol Webster. 2021. Paraphrasing compound nominalizations. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8023–8028, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Catherine Macleod, Ralph Grishman, Adam Meyers, Leslie Barrett, and Ruth Reeves. 1998. Nomlex: A lexicon of nominalizations. In *Proceedings of EURALEX*, volume 98, pages 187–193.

Adam Meyers, Catherine Macleod, Roman Yangarber, Ralph Grishman, Leslie Barrett, and Ruth Reeves. 1998. Using nomlex to produce nominalization patterns for information extraction. In *The Computational Treatment of Nominals*.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*.

Joakim Nivre, Marie-Catherine De Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, et al. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 1659–1666.

Sebastian Padó, Marco Pennacchiotti, and Caroline Sporleder. 2008. Semantic role assignment for event nominalisations by leveraging verbal data. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 665–672. Association for Computational Linguistics.

Sameer Pradhan, Honglin Sun, Wayne Ward, James H Martin, and Dan Jurafsky. 2004. Parsing arguments of nominalizations in english and chinese. In *Proceedings of HLT-NAACL 2004: Short Papers*, pages 141–144. Association for Computational Linguistics.

Karin Kipper Schuler. 2005. *VerbNet: A broad-coverage, comprehensive verb lexicon*. University of Pennsylvania.

Aryeh Tiktinsky, Yoav Goldberg, and Reut Tsarfaty. 2020. pybart: Evidence-based syntactic transformations for ie.

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

Mohamed Yahya, Steven Whang, Rahul Gupta, and Alon Halevy. 2014. Renoun: Fact extraction for nominal attributes. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 325–335.

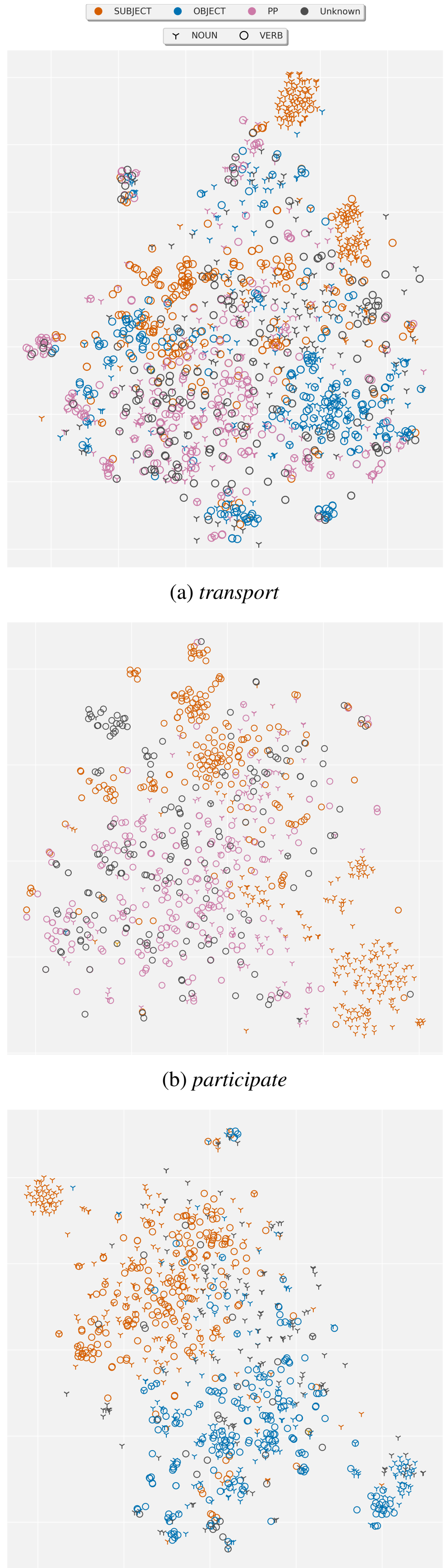
Yanpeng Zhao and Ivan Titov. 2020. Unsupervised transfer of semantic role models from verbal to nominal domain. *arXiv preprint arXiv:2005.00278*.

A Verb-Noun Argument Similarity

The similarity between arguments of verbs and arguments of matching deverbal noun realizations is a prominent requirement of our method. Similarly, Zhao and Titov (2020) exploit verb-noun similarities and base their approach on this assumption. To explore this similarity, we take the verbal and nominal arguments extracted by NomLex of the types SUBJECT, OBJECT, PP, and undetermined (Unknown), embed them using a pre-trained BERT-large-uncased model, and compare their 2-dimensional representations (using t-SNE algorithm (Van der Maaten and Hinton, 2008) for dimension reduction). These representations are illustrated in Figure 2, demonstrating relatively similar representations between arguments of the verbs *transport*, *participate* and *violate* (marked as 'O') and the matching arguments of the corresponding noun forms (marked as 'Y'). More concretely, most nominal argument representations in these illustrations have a nearby verbal argument neighbor with the correct argument type. This similarity establishes the foundation of our work.

B Extended Main Results

We provide here more information regarding our best results. In Table 5, we state the performance of all suggested methods when applied to the tuning sets, similar to our statement in Table 2. Moreover, Table 6 summarizes the number of instances for the most common verbal relations in each test set and the Relation-F1 score of every such relation. As expected, '*nsubj*' and '*dobj*' are the most common relations in the test sets. Other regarded relations are '*nmod:x*' relations and ∅ relations (referring to non-argument identifications or predictions).



(a) *transport*

(b) *participate*

(c) *violate*

Figure 2: Arguments representations of the verbs *transport*, *participate* and *violate* and their matching nouns

Method	Paraphrasing-derived Relation-F1	Paraphrasing-derived Exact-Match	NomLex$_{paraphrasing}$ Relation-F1	NomLex$_{paraphrasing}$ Exact-Match
baseline (NomLex-based)	42.46	11.48	-	-
all-subject	31.62	0.00	39.16	0.00
all-object	34.78	0.00	37.92	0.00
nearest-avg-argument	54.62	21.31	44.96	21.99
k-nearest-argument	**67.21**	**40.98**	**58.16**	**41.84**

Table 5: The best results of the two suggested labelers on the two tuning sets, compared to the baseline process and the naive methods 'all-subject' and 'all-object'.

Relation Type	Paraphrasing-derived Support	Paraphrasing-derived F1	NomLex$_{paraphrasing}$ Support	NomLex$_{paraphrasing}$ F1	NomLex$_{other}$ Support	NomLex$_{other}$ F1
nsubj	151	71.34	1910	62.86	6825	61.01
dobj	202	79.49	2075	63.08	6277	63.50
$\emptyset$	58	9.45	382	13.22	1191	16.09
nmod:to	24	50.00	162	22.11	419	14.11
nmod:with	14	19.35	77	23.92	404	34.60
nmod:for	11	37.04	105	29.12	322	30.36
nmod:from	2	0.00	86	28.28	276	35.06
nmod:in	41	56.52	233	36.90	263	12.34
nmod:as	8	7.41	99	33.90	220	39.05
nmod:on	5	20.00	49	21.65	218	36.70
nmod:into	2	33.33	26	14.46	114	35.90
nmod:against	1	0.00	25	36.00	96	52.57
nmod:over	0	-	12	0.00	76	35.46
nmod:about	1	0.00	0	-	43	37.21
nmod:at	4	18.18	22	4.26	33	10.66
nmod:of	4	0.00	14	0.00	23	5.13
nmod:towards	0	-	0	-	17	51.43
nmod:through	11	0.00	8	17.14	13	21.05
nmod:across	0	-	2	40.00	9	26.09
nmod:due to	0	-	2	22.22	7	6.45
nmod:between	2	0.00	1	0.00	6	0.00
nmod:among	0	-	1	33.33	6	7.41
nmod:along	1	66.67	0	-	5	0.00
nmod:by	8	0.00	2	0.00	2	0.00

Table 6: The support of the most common verbal relations in the test sets, alongside their Relation-F1 score (as 'F1') of our best method ('k-nearest-argument').

ACL 2023 Responsible NLP Checklist

A For every submission:

☑ A1. Did you describe the limitations of your work?
limitations section

☑ A2. Did you discuss any potential risks of your work?
ethics statement section

☑ A3. Do the abstract and introduction summarize the paper's main claims?
abstract section

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

5

☑ B1. Did you cite the creators of artifacts you used?
5

☐ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Not applicable. Left blank.

☐ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Not applicable. Left blank.

☐ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Not applicable. Left blank.

☐ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Not applicable. Left blank.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
6

C ☑ Did you run computational experiments?

7

☒ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
I did not train new models. Only pre-trained models were used.

The Responsible NLP Checklist used at ACL 2023 is adopted from NAACL 2022, with the addition of a question on AI writing assistance.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
7

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
7

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
5

D ☒ **Did you use human annotators (e.g., crowdworkers) or research with human participants?**
Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.