Extending an Event-type Ontology: Adding Verbs and Classes Using Fine-tuned LLMs Suggestions

Jana Straková and **Eva Fučíková** and **Jan Hajič** and **Zdeňka Urešová**
Institute of Formal and Applied Linguistics, Faculty of Mathematics and Physics
Charles University, Prague, Czech Republic
{strakova,fucikova,hajic,uresova}@ufal.mff.cuni.cz

Abstract

In this project, we have investigated the use of advanced machine learning methods, specifically fine-tuned large language models, for pre-annotating data for a lexical extension task, namely adding descriptive words (verbs) to an existing (but incomplete, as of yet) ontology of event types. Several research questions have been focused on, from the investigation of a possible heuristics to provide at least hints to annotators which verbs to include and which are outside the current version of the ontology, to the possible use of the automatic scores to help the annotators to be more efficient in finding a threshold for identifying verbs that cannot be assigned to any existing class and therefore they are to be used as seeds for a new class. We have also carefully examined the correlation of the automatic scores with the human annotation. While the correlation turned out to be strong, its influence on the annotation proper is modest due to its near linearity, even though the mere fact of such pre-annotation leads to relatively short annotation times.

1 Introduction

Annotation of highly-dimensional, voluminous data is expensive, time-consuming and in addition, in case of deep-niche domains, depending on expertly trained specialists, such as linguists or medical experts. Therefore it may be advantageous to organize, prioritize and provide suggestions to guide further annotation efforts efficiently. Especially in a situation with a rich, constantly growing set of classes, such as it is the case with ontologies.

Specifically, given an already partially labeled set of examples with yet unfinished set of classes, classifier based on large language models (LLMs) can be leveraged to navigate the landscape of possible annotations.

Our showcase is an event-type ontology, the SynSemClass 4.0 (Uresova et al., 2022), populated with synonymous verbs denoting events or states. The set of events is currently dynamically evolving and encompasses classes in English, Czech, German and Spanish, so far limited to verbs.

As any ontological resource is never complete, we have investigated various methods to facilitate efficient extension of such ontologies in two ways: adding classes for greater coverage on new texts, and adding verbs to existing classes to allow for more accurate human understanding of the classes in the ontology for a particular form of the given class expression.

We suggest to achieve these by

1. examining examples with consistently low class affiliation scores across a large corpus as potential candidates for new classes;

2. on the other side of the spectrum, examining high-certainty decisions of a supervised classifier to locate highly-affiliated lemmas to a particular class, corresponding to "low-hanging fruit" for a quick manual review and confirmation of the inclusion of the lemma into the suggested class.

In all cases, classifier prediction serves as guidance and the annotators are briefed to consider the suggestions as election votes. The final decision is always the annotator's, who can accept or dismiss the suggestions.

The organization of this paper is as follows: Sect. 2 introduces the SynSemClass v4 ontology and the current state of annotations. Sect. 3 describes the fine-tuned LLM classifier used to generate the annotation suggestions. Sect. 4 describes the manual annotations post-processing. Results are presented in Sect. 5 and discussed in Section 6. Finally, we conclude in Sect. 7.

We release the source code at https://github.com/strakova/synsemclass_ml.

Proceedings of the 17th Linguistic Annotation Workshop (LAW-XVII), pages 85–95

2 The Ontology

In our experiments, we have used the Czech part of the SynSemClass 4.0[1] (Uresova et al., 2022) in which contextually-based synonymous verbs in various languages are classified into multilingual synonym classes according to the semantic and syntactic properties they display. There is no specific model or lexicographic theory behind building the database. However, from the linguistic point of view, the notion of synonymy used is based on the "loose" definition of synonymy by Lyons and Jackson (Lyons, 1968; Jackson, 1988), or alternatively and very closely, on both "near-synonyms" and "partial synonyms" as defined by Lyons (Lyons, 1995; Cruse, 2000) or "plesionyms" as defined by Cruse (Cruse, 1986).[2]

From the ontological point of view, the classes are meant to reflect different event types (concepts) and collect various information about the possible forms of expression of the event type in language.

The following main basic features are distinguished in SynSemClass (Fig. 1) (Uresova et al., 2022):

- The **name of each multilingual class** stands for a single concept (e.g., of *accelerating*)[3] and corresponds to the verb that represents the prototypical sense in each of the languages included: class member (CM) *abuse* for English, *zneužívat* for Czech, and *missbrauchen* for German. So far, SynSemClass focuses on verbal synonyms since they carry the key syntactic-semantic information for language understanding.[4]

- Each class is also provided with a brief language-dependent general **class definition**, which characterizes the meaning, or concept

Figure 1: SynSemClass example entry as presented on its public access website, Class ID: vec00591 (simplified)

[1] https://lindat.mff.cuni.cz/repository/xmlui/handle/11234/1-4746

[2] The "loose" definition of synonymy covers synonyms that fulfil some of the conditions stipulated for synonymy in the strictest sense but not all and does not work with the "absolute" synonymy covering the total identity of meaning. The "partial synonymy" is defined (Lyons, 1995) as a relationship holding between two lexemes that satisfy the criterion of identity of meaning, but do not meet all the conditions of absolute synonymy. The "near-synonymy" (Lyons, 1995) and "plesionyms" (Cruse, 1986) is defined as "expressions that are more or less similar, but not identical, in meaning".

[3] This is different from the commonly used term of "semantic classes of verbs" as represented, for example, in VerbNet, where the class is defined much more broadly – such as for all verbs of movement.

[4] As described in detail in (Urešová et al., 2019, 2018a,c,b).

of the class, i.e. the meaning of all synonymous verbs contained in it. The class is viewed as a substitute for an ontology unit representing a single concept, similar to the treatment of WordNet synsets in (McCrae et al., 2014).

- For each class, SynSemClass also provides a fixed set (called **"Roleset"** (RS)) of defined "situational participants" (called "semantic roles" (SR)) that are common for all the members (the individual verb senses) of a particular class. The RS is mapped to the valency frame of the individual synonymous verbs securing for each synonymous verb to be characterized both meaning-wise (SR) and structurally (valency arguments). For example, the class `vec00591` *abuse*, as concept of "abusing", has two semantic roles, `Abuser` and `Abused` (Fig. 1). Every role in SynSemClass is provided with a brief language-dependent general **role definition** as well as every class. While the SRs resemble FrameNet's "Frame Elements" (and sometimes borrow their names from there), it should be pointed out that there is one fundamental difference: the SRs used in SynSemClass aim at being defined across the ontology, and not per class (as they would be if we follow the "per frame" approach used in FrameNet).

- Each individual language-dependent synonymous verb included in a given class is called **Class Member** and for each new CM to be added, it must be possible, in the prototypical case, to create a mapping between its syntactic arguments and the roles in that class' RoleSet; see the example in our web-based lexicon (Fig. 1).[5] Each CM of one class is denoted by a verb lemma and the valency frame ID which, roughly speaking, represents the particular verb sense.

- Each CM is further linked to one, or more existing online lexical resources for each language to support, e.g., comparative studies, or any other possible research in the community. In SynSemClass (SSC), there exist **links** to e.g., Vallex[6] for Czech, FrameNet[7] and VerbNet[8] for English, E-VALBU[9] for German, AnCora[10] for Spanish. Each Class Member is exemplified by instances of real texts (and their translations to English) extracted from translated or parallel corpora. Specifically, data is extracted from the Prague Czech-English Dependency Corpus (PCEDT)[11] for Czech-English, from the Paracrawl corpus[12] for German-English and from the XSRL dataset[13] for Spanish-English.

SynSemClass 4.0 includes 1200 classes (885 active after merging or deleting) with 8169 Class Members. All classes are annotated in Czech and English, 60 of them have also German annotation. Spanish is not included in the web version but is under construction (Fernández-Alcaina et al., 2023).

3 Generating Annotation Suggestions with Fine-tuned LLM Classifier

3.1 Data

The Czech part of the SynSemClass ontology[14] yielded 12045 example sentences with 3313 unique verbs (lemmas) manually annotated in 965 classes.[15] We have split the data randomly in proportion 80/10/10 in a stratified train/dev/test split,[16] resulting in 9635/1205/1205 train/dev/test examples.

Our input is a list of 3389 completely new, unseen verbs (lemmas) and our motivation is to differentiate:

- verbs consistently poorly classified as class members of any of the existing classes, i.e., possible candidates for establishing new classes,

[5] The public web version is available at https://lindat.cz/services/SynSemClass40/SynSemClass40.html

[6] https://hdl.handle.net/11234/1-3524

[7] https://framenet.icsi.berkeley.edu/fndrupal/

[8] https://uvi.colorado.edu/andhttp://verbs.colorado.edu/verbnet/index.html

[9] https://grammis.ids-mannheim.de/verbvalenz

[10] http://clic.ub.edu/corpus/es/ancoraverb_es

[11] https://ufal.mff.cuni.cz/pcedt2.0/en/index.html

[12] https://opus.nlpl.eu/ParaCrawl.php

[13] https://catalog.ldc.upenn.edu/LDC2021T09

[14] SynSemClass 4.0 with additions annotated since the last v4.0 release.

[15] We considered only active (not merged, not deleted) classes in the current state of SynSemClass (SSC) annotated since v4.0 release, and naturally, only those classes which are represented with at least one example sentence (to be used as LLM input).

[16] Stratified means forcing the distribution of the target variable, in our case the classes, to be equal among the train/dev/test split.

- verbs highly affiliate to some of the existing classes, i.e., possible candidates for adding them as one of the verbs characterizing an existing class.

To obtain the classification score for each lemma-class pair, we used a large raw corpus of written Czech, the SYN v4 (Křen et al., 2016; Hnátková et al., 2014).[17] Specifically, we used the first 2.753.494 sentences of the corpus, which amounts to exactly 100-th of all its sentences, as classifying the corpus in its entirety (275.349.474 sentences) is above our GPU computation means. The classification took 20 hours on a single NVIDIA A100 GPU with 4 CPU threads.

3.2 Model

Classification tasks on a finalized set of target variables are usually modeled as a probability distribution over K targets (possible outcomes). However, we find ourselves in an untypical situation in which the output target set is not closed yet, which requires a different perspective. If we model the problem as multi-class probability distribution, we will face an out-of-distribution problem concerning verbs which do not belong to any of the classes. We therefore model the problem as K independent binary classifiers, one for each class, of which each predicts the probability of the input belonging to the particular class in question, much like a multi-label problem. Technically, this equals to replacing the output softmax activaction function with the sigmoid activation function and accommodating the loss function accordingly, from sparse categorical cross entropy to sparse binary (focal) cross entropy,[18] while the weights are estimated jointly by fine-tuning one shared large language model.

3.3 Training

Our classifier is a fine-tuned RemBERT (Chung et al., 2021), a rebalanced 559M-parameter mBERT,[19] with sigmoid activation function on the output layer and sparse binary focal cross entropy ($\gamma = 2.0$) to model the target class probabilities independently (see also previous Section 3.2). We trained our model using the Adam optimizer (Kingma and Ba, 2015) with defaults β's and with a batch size of 10. The model was fine-tuned on a single NVIDA A100 GPU, using linear warm-up in the first training epoch (6.66% training steps) from 0 to peak learning rate $1 \cdot 10^{-5}$ and then decaying with a cosine decay schedule (Loshchilov and Hutter, 2017). The model was trained for 15 epochs and we used dropout with probability 0.5. The hyperparameters were tuned on the development set; the model achieved development set accuracy 78.67% and test set accuracy 79.17%.

3.4 Related Work

We are not aware of a similar work using LLMs to classify words (and specifically, verbs) into synonym classes to enrich an existing ontology or lexicon. There are works building such resources from scratch, starting from (Brown et al., 1992) the model and its statistical, unsupervised class hierarchy building algorithm.

The ASFALDA project ("Analyzing Semantics with Frames: Annotation, Lexicon, Discourse and Automation")[20] aims at projecting English FrameNet frames to French also using machine learning but it is a recently started project and there are no published results yet.

The Predicate Matrix project (Lopez de Lacalle et al., 2016) aims at creating a resource similar to SynSemClass, by using similar resources that SynSemClass links to. The entries created automatically are not manually checked (for the most part) and we are not aware of publications describing if there were specific experiments on the comparison of the automatically created entries vs. human annotation.

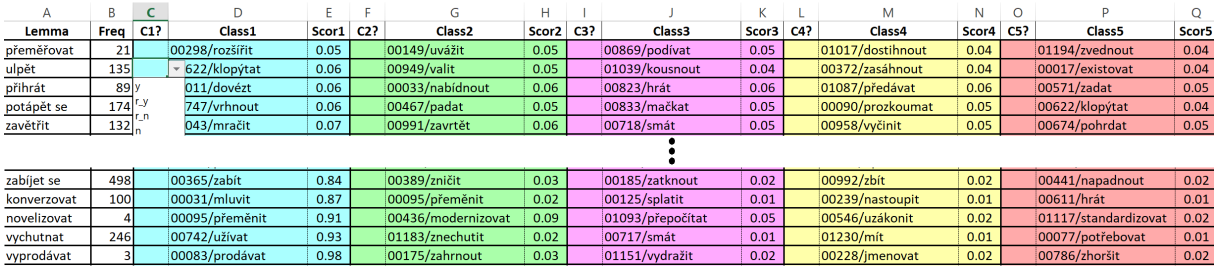
There is also work on using DNNs (LSTMs specifically) to model lexical ambiguity (Aina et al., 2019), which is relevant for our task, but the method is not related to another existing ontological or lexical resource for training and/or fine/tuning the ML part of the system.

[17] http://hdl.handle.net/11234/1-1846

[18] "Focal" stands for focal loss (Lin et al., 2018), which addresses class imbalances in training data by encouraging learning on the sparse set of hard examples (the rare positives in our case, because only one of hundreds of classes is correct) and discouraging learning from a vast majority of easy (negative) examples.

[19] Although BERT (110M parameters) and RemBERT (∼0.5B parameters) are technically considered large language models (LLMs), they certainly rank among the modest language models w.r.t. number of parameters. Quite precisely, they belong to the masked language models (MLMs) family. Our method can however be used with any fine-tuned LLM.

[20] https://anr.fr/Project-ANR-12-CORD-0023



A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
Lemma	Freq	C1?	Class1	Scor1	C2?	Class2	Scor2	C3?	Class3	Scor3	C4?	Class4	Scor4	C5?	Class5	Scor5
přeměřovat	21		00298/rozšířit	0.05		00149/uvážit	0.05		00869/podívat	0.05		01017/dostihnout	0.04		01194/zvednout	0.04
ulpět	135		622/klopýtat	0.06		00949/valit	0.05		01039/kousnout	0.04		00372/zasáhnout	0.04		00017/existovat	0.04
přihrát	89	y	011/dovézt	0.06		00033/nabídnout	0.06		00823/hrát	0.06		01087/předávat	0.06		00571/zadat	0.05
potápět se	174	r_y	747/vrhnout	0.06		00467/padat	0.05		00833/mačkat	0.05		00090/prozkoumat	0.05		00622/klopýtat	0.04
zavětřit	132	r_n n	043/mračit	0.07		00991/zavrtět	0.06		00718/smát	0.05		00958/vyčinit	0.05		00674/pohrdat	0.05
zabíjet se	498		00365/zabít	0.84		00389/zničit	0.03		00185/zatknout	0.02		00992/zbít	0.02		00441/napadnout	0.02
konverzovat	100		00031/mluvit	0.87		00095/přeměnit	0.02		00125/splatit	0.01		00239/nastoupit	0.01		00611/hrát	0.01
novelizovat	4		00095/přeměnit	0.91		00436/modernizovat	0.09		01093/přepočítat	0.05		00546/uzákonit	0.02		01117/standardizovat	0.02
vychutnat	246		00742/užívat	0.93		01183/znechutit	0.02		00717/smát	0.01		01230/mít	0.01		00077/potřebovat	0.01
vyprodávat	3		00083/prodávat	0.98		00175/zahrnout	0.03		01151/vydražit	0.02		00228/jmenovat	0.02		00786/zhoršit	0.02

Figure 2: Preprocessed data with 5 suggested classes per lemma, first and last five lines, as presented to the annotators in an Excel spreadsheet (the version with scores shown; cursor (in column C, line 3, 2^{nd} data line) shows the annotation choices)

4 Post-processing with Manual Annotations

4.1 Input Data Preparation

In the output of the automatic classifier, each lemma has been associated with ten highest-scoring classes in which the lemma can potentially be inserted as a class member. The score is thus assigned to each lemma-class pair. These scores are numbers between 0 and 1, but it is not a probability but really just a "score" or a "weight." The smaller the score, the less is the used LLM sure that the verb lemma belongs to the class, and vice versa - the higher the score, the more convinced it is to be added to the class.

The data as received from the classifier (3073 lemmas, with 10 class suggestions and scores for each of them) have been converted to an Excel spreadsheet to be presented to the annotators as follows:

1. For each lemma (line in the resulting file), the first five classes suggested by the classifier with the highest scores as assigned by the classifier have been kept;

2. two disjunct sets of lemmas and their class membership suggestions, with 100 lemmas each, have been randomly selected from the 3073 lemmas scored by the classifier;

3. the two sets (called Set1 and Set2) have been converted to an Excel spreadsheet, keeping frequency information for the lemma, the five highest-scoring class membership suggestions, and the associated scores with each class;

4. in front of each class suggestion, an extra column has been inserted with the four-way list of decisions the annotators will have to make;

5. colors have been used to group all the information pertaining to one lemma-class pair and the decision requested;

6. for each class suggested, a web link has been inserted in its spreadsheet cell, to allow the annotator to get to the class definition and contents (which is available on the web as shown in Fig. 1) by a single click.

Then, each set has been duplicated and in the copy, the scores have been deleted. The four files have then been renamed to contain the annotator abbreviation and the order number (1 for the version without scores, 2 for the version with scores (see Fig. 2), i.e., in a cross-named way for the Set1 and Set2; see Table 1).

Annotator:	A1	A2
1^{st} batch (no scores shown)	Set1	Set2
2^{nd} batch (scores shown)	Set2	Set1

Table 1: Order and Assignment of Data to Annotators

4.2 Experiment Design

The Excel spreadsheets as described in the previous section (Sect. 4.1) have been sent to two annotators in two batches: first, both received the file with five suggestions for each lemma, but no scores. Each thus had 500 decisions to make (100 lemmas × 5 classifier suggestions per lemma) on a four-point scale, 0-3, denoting how strongly they recommend to include the lemma in the suggested class. The "no" decision corresponds to 0, "rather_no" to 1, "rather_yes" to 2, and "yes" to 3. These responses have been provided in the Excel spreadsheet as a fixed list, in order to avoid typos. In the second batch, the annotators received the other 100 lemmas, this time with scores denoting the classifier's

view on the strength of the class membership recommendation, for the five classes presented.

In total, there were thus 200 lemmas manually classified twice (by the two annotators), with the classifier scores shown only for half of them to each annotator. No annotator annotated any lemma twice, and they worked independently without consulting each other. The annotators, native speakers of Czech, have been previously trained on the same task (with data coming from a different preprocessing method), so no additional training has been performed. Their pay has been based on hours worked, approx. $8/hour amounting to about 170% of the legal minimal salary valid in 2023 in the Czech Republic.

The order and cross-assignment of the data to the annotators allowed us to measure interannotator agreement and the correlation between the annotators decisions (averaged) and the automatic classifier recommendations. Also, we could compare the speed of annotation with and without the additional clue, namely, the scores suggested by the automatic classifier.

5 Results

This section describes the results obtained as described in Sect. 3 and Sect. 4. For the discussion of the various outputs, see Sect. 6.

5.1 Human Annotation Statistics and IAA

There were 1000 pairs of Czech verb and suggested class in two sets (Set1 and Set2, see Sect. 4.1). The two annotators, A1 and A2, had to decide whether the verb could be a member of the class. Annotators could set 4 values: "yes," "rather_yes," "rather_no" or "no,". Agreement was calculated for only two values, Y and N, to which the four detailed levels of annotation have been mapped in a natural way (specifically, "rather_no" has been mapped to "N" and "rather_yes" to "Y"). The (dis)agreement figures have been counted based on each individual decision as made by the two annotators. The resulting counts are shown in the Tab. 2 and agreement rate and Cohen's κ value in the Tab. 3.

5.2 Human Annotation Time

The annotators have been asked to record the time it took them to annotate the data. Each Set entailed 500 decisions, which took slightly over three hours on average. The detailed breakdown is shown in Tab. 4.

A1\A2	Y	N	Total
Y	129=66+63	43=15+28	172=81+91
N	122=57+65	706=362+444	828=419+409
Total	251=123+128	749=377+372	1000=500+500

Table 2: Annotation statistics: counts shown for the 1000 annotation decisions (500 from Set1, 500 from Set2). Mappings used: y $\rightarrow$ Y, r_y $\rightarrow$ Y, r_n $\rightarrow$ N, n $\rightarrow$ N. Counts are presented in the cells as Total-xy=Set1-xy+Set2-xy, where $x, y \in \{$Y, N$\}$.

	IAA	Cohen's κ
All	0.83	0.51
Set1	0.86	0.56
Set2	0.81	0.46

Table 3: Inter-annotator agreement and Cohen's κ between annotators, for the 500 decisions each annotated by both annotators, with the scaled values mapped to Y/N only.

	Batch 1 (no scores)	Batch 2 (with scores)
A1	192	174
A2	210	210
Average	201	192

Table 4: Time of annotation by annotators A1 and A2, in minutes. Batch 1 is Set1 and Set2 without showing the scores assigned by the automatic classifier, Batch 2 shows the scores.

5.3 Correlation between the Scores of the Automatic Classifier and the Human Annotation

To find if there is a relationship between the automatic scores and manually annotated data, we used the Pearson's correlation (Pearson's r) coefficient. Automatic scores and human annotations were found to be moderately correlated ($r(998) = .44$, $p < .001$). A Spearman's correlation was also run to determine the relationship between 1000 automatic scores and human annotations. There was weak to moderate monotonic correlation between automatic scores and human annotations ($\rho = .39$, $n = 1000$, $p < .001$).

We visualize the correlation between the automatic scores assigned to the lemma-class pairs and annotation decisions in Fig. 3; human scores correspond to the annotation scale (3 - yes, 2 - rather_yes, 1 - rather_no and 0 - no) and automatic scores are bucketed (interval size: 0.05) and annotation decisions averaged in each bucket, effectively smoothing out the curve by reducing variance. The Pearson correlation between scores and human

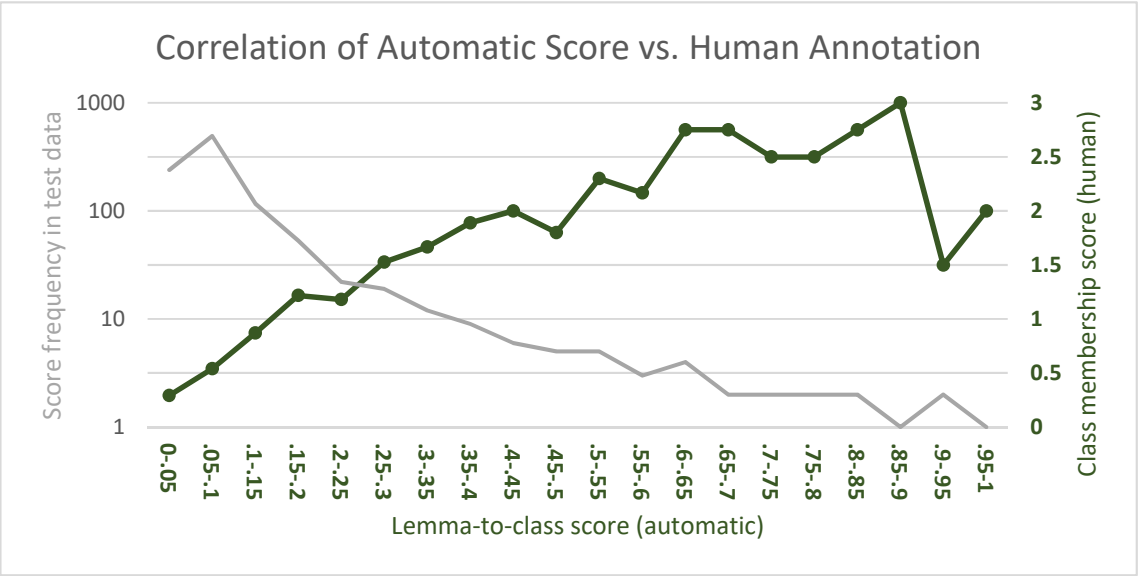


Figure 3: Correlation between the automatic scores assigned to the lemma-class pairs and annotation decisions; human scores correspond to the annotation scale (3 - yes, 2 - rather_yes, 1 - rather_no and 0 - no) and automatic scores are bucketed (interval size: 0.05) and annotation decisions averaged in each bucket. The grey line shows the size of each bucket on a logarithmic scale.

annotations averaged per bucket is $r(18) = .79$ ($p < .001$) and Spearman's ranked correlation is $\rho = .76$ ($n = 20, p < .001$).

6 Discussion of Results

It is well known that trained annotators often create high-quality data, needed for many NLP applications, although their services are generally expensive.The experiment described here was designed to answer several questions:

- What is the usual inter-annotator agreement for the human assignment of verbs to classes, using pre-annotated data?
- Can a heuristics be defined to indicate which pre-assigned lemma-class pairs the annotators can trust and to what extent?
- Does the scoring mechanism, which provides scores for each of the lemma-class relation strength, make the annotation more efficient?
- Is the automatic classifier for computing the relation strength between an unknown lemma and an existing class(es), as described in Sect. 3, in any way correlated with the human decisions made by experienced annotators?

As seen from Sect. 5.1 (Tab. 3), the inter-annotator agreement is relatively high (0.83 on average over the two Sets), but the Cohen's kappa κ is low (0.51 on average over the same two Sets annotated). However, the low kappa is caused by the highly skewed distribution of the decisions,[21] the most of which lead to the rejection of the assignment of the lemma to the suggested class, caused mainly by the selection of a fixed number of five suggestions per lemma regardless of the score computed by the classifier. It would be possible - by using more pairs of annotators - to optimally select the number of suggested classes (i.e., most probably between 1 and 5), but it would only be relevant for the current number of classes in the ontology. As the ontology grows, the number of rejections will be different and the optimal number of classes might change.

For the size of the ontology on which it has been tested, the threshold separating the Yes/No decision (with the highest uncertainty being around the average of 0-3, i.e., 1.5) seems to be around 0.3 (see Fig. 3). However, due to the linearity of the correlation (which by itself is a positive result for the classifier–see below), it would still be necessary to provide careful manual inspection results on both sides of the threshold. The same holds for setting any thresholds at the low or high ends of the classifier score scale. In terms of annotation efficiency (providing scores to the annotators vs.

[21] Almost 4:1 - the average number of (mapped) "No" decisions is 788,5 out of 1000.

not providing them), the result is largely negative. A small speedup has been observed only for A1, with A2 consuming the same time for both Sets. The absolute time as recorded by the annotators per lemma (i.e., 5 times the single decisions time, which was $366 \times 60 \div 1000 \approx 22$ sec. for A1 and $420 \div 1000 \approx 25$ sec. for A2) is about two minutes. This is in fact a positive finding which means that the whole set of pre-classified lemmas, as processed by the classifier (3073 lemmas) would be finished within approx. 6000 minutes (100 hours) per annotator, i.e., within 200 hours with double annotation, plus adjudication time.

Finally, the correlation between the automatic classifier and the human annotation is very strong. Of course, the bucketing to the 0.05 interval improves the correlation (see Sect. 5.3), but in any case, it seems that the classifier is able to assign the score denoting the strength of affiliation of the unknown lemma to a class with high correlation to the human annotation decisions.

7 Conclusions and Future Work

As discussed in the previous section (Sect. 6), the strongest result achieved in this study is the correlation between the classifier scoring buckets and the human decisions (Fig. 3, Sect. 5.3). While the scores themselves, when presented to the annotators, do not seem to bring higher efficiency, the selection of the classes and their presentation to the annotators (Sect. 3, Sect. 4.1) result in a reasonable time for the annotation of several thousand previously unseen (unassigned lemmas) to the ontology. Finally, there is no strong heuristics (for the score thresholds) that would allow to assign any unseen words to existing classes automatically – a human post-inspection and annotation is needed across the whole (or almost whole) range of scores as produced by the classifier, given the linear correlation.

In the future, we plan to repeat the experiment for a larger ontology (i.e., test the effort needed for sustainable development and maintenance for such an event-type ontology when it reaches high coverage), possibly with larger LMs or with some additional fine tuning given the large(r) coverage at such future time.

Limitations

We advocate for a moderate and restrained usage of automatic guiding methods and we must advise caution to take the automatic output with a grain of salt, both qualitatively and quantitatively. First, the classifier predictions can fall far from gold labels and should not be considered as such. Second, although measures have been taken to mitigate the out-of-distribution classification problem, one should be aware of the fact that by the very nature of the problem, which is annotation of completely new, possibly out-of-distribution data, the classification predictions are not to be trusted indiscriminately and should subsequently be approved by the annotators. The annotators should be instructed to consider the suggestions as election votes. Furthermore, we should refrain from overly automating the entire annotation process so as to achieve high alignment with the machine learning suggestions, which might lead to trivial and unimaginative annotations from the linguistic perspective. Finally, exhausting the informativeness of the pre-trained (albeit fine-tuned) model might prevent further learning from the annotated data.

Another limitation of the results, or the interpretation of the results, is the fact that the model is trained on an actual state of the ontology. It means that in fact the classifier would have to be retrained after adding a single new class or even a new lemma to an existing class; while in practice it would be OK to process several lemmas at once, it is still a limitation given the non-negligible training and prediction time (20 hours on a single GPU) which cannot be parallelized (see Sect. 3).

In addition, the correlation might decrease and the thresholds shift as the size (and thus coverage) by the ontology grows, since the unseen lemmas will be increasingly rare, with possibly less data available in the LM to reliably estimate the scores. Conversely, for ontologies with much smaller coverage (e.g., for ontologies the development of which has just started) the same shifts in correlation and thresholds are likely.

Finally, the whole experiment has been performed on Czech due to the lower coverage of the ontology than for English, and also in order to explore a morphologically rich language with a high form-to-lemma ratio. Results for other languages might differ.

Ethics Statement

The human subjects used in this study have been experienced, trained annotators who have been in personal contact with the authors, and who have been recruited by a call specifically suited for the

experiment and study presented here. The call has been sent to all trained annotators already working with the authors, and volunteers have been asked to respond, on a first-come first-chosen basis. The pay has corresponded to the standard pay for similar annotation tasks taking also the relatively short notice into consideration (for the numbers, see Sect. 4.2). Both annotators were males; this is a possible shortcoming, but there were no female volunteers and from the previous cooperation (with a mixed team of female and male annotators), no differences in the annotation results have been observed.

No personal information has been among the lemmas extracted and used for the preselection. The data for the LLM might have contained it, but it would not show because the experiment and the ontology is currently limited to common verbs which do not describe any personal names or other personal information.

Acknowledgements

The work described herein has been supported by the Grant Agency of the Czech Republic under the EXPRO program as project "LUSyD" (project No. GX20-16819X) and uses resources hosted by the LINDAT/CLARIAH-CZ Research Infrastructure (projects LM2018101 and LM2023062, supported by the Ministry of Education, Youth and Sports of the Czech Republic).

We would like to thank the annotators Petr Kujal and Tomáš Razím for their valuable work and invaluable input, as well as the three anonymous reviewers for their insightful comments.

References

Laura Aina, Kristina Gulordava, and Gemma Boleda. 2019. Putting words in context: LSTM language models and lexical ambiguity. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3342–3348, Florence, Italy. Association for Computational Linguistics.

Peter F. Brown, Vincent J. Della Pietra, Peter V. deSouza, Jenifer C. Lai, and Robert L. Mercer. 1992. Class-based *n*-gram models of natural language. *Computational Linguistics*, 18(4):467–480.

Hyung Won Chung, Thibault Fevry, Henry Tsai, Melvin Johnson, and Sebastian Ruder. 2021. Rethinking embedding coupling in pre-trained language models. In *International Conference on Learning Representations*.

Alan Cruse. 2000. *Meaning in Language. An Introduction to Semantics and Pragmatics*. Oxford University Press. Oxford, UK.

D.Alan Cruse. 1986. *Lexical Semantics*. Cambridge University Press, UK.

Cristina Fernández-Alcaina, Eva Fučíková, and Zdeňka Urešová. 2023. Spanish verbal synonyms in the synsemclass ontology. In *Proceedings of the 21st International Workshop on Treebanks and Linguistic Theories*, pages 10–20, Washington, D.C., USA. Association for Computational Linguistics, Association for Computational Linguistics.

Milena Hnátková, Michal Křen, Pavel Procházka, and Hana Skoumalová. 2014. The SYN-series corpora of written Czech. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 160–164, Reykjavik, Iceland. European Language Resources Association (ELRA).

Howard Jackson. 1988. *Words and Their Meaning*. Routledge.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *ICLR (Poster), CoRR abs/1412.6980*.

Michal Křen, Václav Cvrček, Tomáš Čapka, Anna Čermáková, Milena Hnátková, Lucie Chlumská, Tomáš Jelínek, Dominika Kováříková, Vladimír Petkevič, Pavel Procházka, Hana Skoumalová, Michal Škrabal, Petr Truneček, Pavel Vondřička, and Adrian Zasina. 2016. SYN v4: large corpus of written czech. LINDAT/CLARIAH-CZ digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Tsung-Yi Lin, Priyal Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. 2018. Focal loss for dense object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP:1–1.

Maddalen Lopez de Lacalle, Egoitz Laparra, Itziar Aldabe, and German Rigau. 2016. A multilingual predicate matrix. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 2662–2668, Portorož, Slovenia. European Language Resources Association (ELRA).

Ilya Loshchilov and Frank Hutter. 2017. SGDR: Stochastic gradient descent with warm restarts. In *International Conference on Learning Representations*.

John Lyons. 1968. *Introduction to Theoretical Linguistics*. Cambridge University Press.

John Lyons. 1995. *Linguistic Semantics*. Cambridge University Press.

John P. McCrae, Christiane Fellbaum, and Philipp Cimiano. 2014. Publishing and Linking WordNet using lemon and RDF. In *Proceedings of the 3rd Workshop*

on Linked Data in Linguistics, colocated with LREC 2014, Reykjavik, Iceland.

Zdeňka Urešová, Eva Fučíková, Eva Hajičová, and Jan Hajič. 2018a. Creating a Verb Synonym Lexicon Based on a Parallel Corpus. In *Proceedings of the 11th International Conference on Language Resources and Evaluation (LREC'18)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Zdeňka Urešová, Eva Fučíková, Eva Hajičová, and Jan Hajič. 2018b. A Cross-lingual synonym classes lexicon. *Prace Filologiczne*, LXXII:405–418.

Zdeňka Urešová, Eva Fučíková, Eva Hajičová, and Jan Hajič. 2018c. Defining verbal synonyms: between syntax and semantics. In *Proceedings of the 17th International Workshop on Treebanks and Linguistic Theories (TLT 2018), Vol. 155*, Linköping Electronic Conference Proceedings, pages 75–90, Linköping, Sweden. Universitetet i Oslo, Linköping University Electronic Press.

Zdeňka Urešová, Eva Fučíková, Eva Hajičová, and Jan Hajič. 2019. Meaning and Semantic Roles in CzEngClass Lexicon. *Jazykovedný časopis / Journal of Linguistics*, 70(2):403–411.

Zdenka Uresova, Karolina Zaczynska, Peter Bourgonje, Eva Fučíková, Georg Rehm, and Jan Hajic. 2022. Making a semantic event-type ontology multilingual. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 1332–1343, Marseille, France. European Language Resources Association.

Appendices

Classifier and Annotation Results

We are providing Supplemental material with the raw classifier file and the human annotation results. The open-source code and the data itself are provided at GitHub (https://github.com/strakova/synsemclass_ml). Here, technical description of the supplemental material is provided on top of what has been mentioned in the paper.

Classifier output

The raw output of the classifier, with the 3073 previously unseen (unassigned) lemmas and their classification scores to 10 closest classes, is attached in the Supplemental material file (file `all_buckets_2753494.txt`).

The file contents is structured as follows (each lemma and classifier scores are on a single line):

```
lemma freq-in-data max-score suggested-class-1 score-class-1 ... suggested-class-10 score-class-10
```

where

`lemma`
: is the lemma which has been classified to all the available classes in SynSemClass

`freq-in-data`
: is the frequency of the lemma in the dataset used for building the LM

`max-score`
: is the maximum score (score of the first class in the list)

`suggested-class-n`
: is the ID and name (& Czech sense ID) of the n-th best class assigned to the `lemma` by the classifier

`score-class-n`
: is the score assigned to the (`lemma`,`suggested-class-n`) pair.

Annotation Results

The annotation results are presented as four Excel Spreadsheets, named `law-Am-n.xlsx`, where `m` is the annotator ID and `n` is the batch number (i.e., the lemmas and classes are identical for `A1-1` and `A2-2` and for `A1-2` and `A2-1`, except for the presence of scores and differing also of course in the assigned `y/r_y/r_n/n` labels by the annotators).

Each Excel file has 100 content lines (100 lemmas and 5 best classes for each as classified by the pre-annotation tool):

`Lemma`
: is the lemma being classified

`Freq`
: is the (informative-only) frequency of the lemma in the training text

`Cn?`
: is the column where the annotators recorded their decisions

`Classn`
: is the ID of the class (clickable)

`Scorn`
: is the score of the lemma-class affiliation by the classifier (in ...-2.xlsx files only)

`AnnotatorComment`
: is an optional annotator's comment.