Results of the project FK 125217
Distributional models of lexical knowledge

When submitting the research proposal for this project, we planned to use word embedding models, which, at that time could be considered a state-of-the-art method providing a novel way to handle distributional semantics. Our project had a planned timespan of four years. Finally, it ended two years later than originally planned: the first project year was followed by a one-year-long interruption due to the birth of a child of the PI, and another one-year extension at the end of the project was the result of the COVID pandemic. In the six years that passed since the start of our project, enormous advances took place in language technology, making the models we originally planned to apply and the goals we originally set up outdated. We thus needed to constantly change and refine our goals and methods.

In the first year of the research, **we created lexical resources** characterizing features of nominal and adverbial elements as well as of slots in verbal argument frames relevant to the task of identifying instances of the given construction. We defined identification of the semantics of grammatical constructions as a task of generating meaningful questions concerning the given instance of the construction (e.g. a locative vs. oblique interpretation of a specific occurrence of phrase ending in a locative suffix can be characterized by the question adequate to the occurrence). We thus **used questions as criteria** to test whether a meaning distinction is relevant, or a specific semantic characterization of a construction is adequate.

From the point of view of generating questions concerning non-predicative noun phrases (who vs. what), distinctions like person vs. thing or even groups or organizations are relevant. Thus, we created a semantic categorization of nouns based on these criteria. However, since noun phrases are also used predicatively and, in that case, a more fine-grained classification is needed, we further elaborated our initial classification introducing classes like profession, animal, tool, behaviour, etc. This latter classification, by the way, makes it possible to generate more specific questions even for non-predicatively used noun phrases, such as "What animal did you see in the garden?" vs. "What did you see in the garden?".

Concerning adverbially used phrases, we concentrated on the distinction of locatives vs. oblique use of the same case endings and lexicalized adverbial constructions. We identified lexical items that typically occur as specific types of adjuncts.

On the other hand, we also **created of a verbal argument frame lexicon** in which we used thematic roles as the key elements that determine what questions can be generated concerning the predicate itself with the given argument as an anchor. E.g. "What did John do to Jack?" is an appropriate question if John is an agent and Jack is a patient. In the same situation, "What happened to Jack?" is another appropriate question.

We applied the same methodology to phrases containing the instrumental case distinguishing tool usage, sprayed/spread moving patients, agents acting together, having something as a constituent or a feature, using a vehicle, being in a state, wearing something as a garment and various other uses of this case ending.

In the whole process, we used word embedding models to identify and cluster the relevant lexical items. The word clusters could be used as a good starting point for the manual annotation process. As

for the annotation of the predicate frames, we **also distinguished light verb constructions**, where it makes no point to ask a question about the nominal element at the core of the construction, e.g. the question "What did you make?" is odd for the sentence "I made a decision."

We categorized
- the 5000 most frequent words in the instrumental case,
- the 2000 most frequent lexical items that have an instrumental dependent,
- the 13500 most frequent adverbs of manner and nouns with a locative suffix.

We listed all relevant argument frames of the most frequent 2000 Hungarian verbs, including light verb constructions of 720 verbs.

The first project year was followed by a one-year interruption of both projects due to the birth of the PI's first daughter.

In the second year, we evaluated character-n-gram-based embedding models (implemented in the fastText library) comparing them to word-based models used in the first year of the project. These models can assign meaningful representations to words never seen in the training data. The price to pay for this improved recall is in some cases a slight blurring of the representation of unrelated words the form of which highly overlap (e.g. darab 'piece' vs. darabont 'infantry soldier') especially in the case of limited training data. We also **created a modified word embedding training algorithm** that makes it possible to create embeddings from annotated text that yields a shared representation of surface word forms and annotations (including e.g. lemma+POS lexemes and dependency relations like 'object of the verb eat'). Simple filtered nearest neighbor queries on the shared representation make it possible to have the model answer complex questions like 'what else do we do to things that we eat' in a meaningful manner.

In the context of **text categorization**, we also **examined the effect of subword tokenization** (e.g. the Sentence Piece model), and found that, surprisingly, using subword tokenization can significantly improve the performance of even character-n-gram-based models by significantly boosting the recall of category labels while only slightly impacting precision. To further improve text categorization performance, we **created a graphical tool to normalize thematic/semantic label sets**. It was used to merge equivalent labels in a label set created manually by authors of news articles as well as for Thing Recognizer features. The tool presents a 2D map of labels based on their embedding vectors. Potentially equivalent vectors are mapped near each other and can be merged moving them on each other.

We also **performed experiments on using semantic** Thing Recognizer **features in syntactic parsers** (the SVM-based Mate dependency parser as well as neural parsers) to improve performance. We achieved some performance gains with the Mate parser. However, this approach did not result in significant gains in performance for the more recent neural parser models. The latter utilize the embeddings themselves (and the distributional semantic knowledge embodied in them), so converting the dense vector semantic representation into a symbolic one is not necessary for the model to utilize them.

We proceeded to create an algorithm that matches the argument frames against actual corpus occurrences. This involves performing appropriate argument frame transformations when matching non-finite verb forms such as infinitives and participles. We used the Hungarian Universal Dependencies (HUD) corpus for this experiment (derived form a small subcorpus of the Szeged Dependency Treebank), as it contains syntactic annotation that is more-or-less compatible with the annotation for numerous other languages present in the HUD corpus. Our goal was to create a model

that can generate a dependency-style annotation with a more fine-grained and semantically enriched annotation set (containing thematic roles) than those present in the original HUD corpus by training a state-of-the art dependency parser on the enriched annotation. However, the Hungarian UD corpus turned out to be too small for this purpose: the whole corpus contains only 1800 sentences (42000 tokens), that is partitioned into 2:1:1 train:dev:test sets. This amount of data turned out to be inadequate to achieve the performance we sought.

We thus undertook to convert the whole Szeged Dependency Treebank (SZDT) (consisting of about 82000 sentences, 1.5 M tokens) into UD format. We needed to solve several problems with SZDT in the process:

1. Orthographic errors are not annotated morphologically (only marked as "erroneous" without further analysis) – this affects more than 10% of the sentences in the corpus
2. Morphological analysis and lemmatization of a major part of verb forms (especially participles) is inadequate
3. The analysis of many (mainly pronominal) elements in SZDT is incompatible with UD annotation principles
4. Even the basic UD dependency set is much more fine-grained than that in SZDT (several one-to-many mappings)
5. The UD head selection principles are quite different from that of the SZDT annotation (including manually introduced zero copulas in SZDT)
6. Subject-predicate annotation is wrong in SZDT for most sentences that involve 3rd person subjects, focus and nominal predication.

We a) corrected spelling errors (this involved merging/splitting/inserting/deleting tokens), mapped the tokens in the corrected sentences to the original tokenized representations, morphologically analyzed and automatically disambiguated the corrected words, b) reanalyzed inadequately analyzed verb forms, and pronominal elements, c) automatically mapped SZDT dependencies to UD disambiguating one-to-many mappings (e.g. the SZDT ATT relation maps to UD amod, nummod, nmod, ccomp, csubj, acl, advcl, case and subtypes thereof depending on context), and converting head-dependent structures where the two annotation schemes apply different principles.

At the time of submitting the research proposal, static word embeddings were used in state-of-the-art NLT solutions. By **the third year of the project, contextual language models** and generative models based on the transformer architecture became ubiquitous and provided end-to-end solutions with nearly human-like performance to many natural language understanding and generation tasks. The most appealing feature of these models from our point of view was that transformer-based models trained on multilingual training data (e. g. multilingual BERT, XLM-RoBERTa) fine-tuned for a specific task in one language, can be utilized to perform that task in another language yielding quite acceptable performance for some tasks (this is called **zero-shot cross-lingual transfer**). This development has inspired us to modify our research agenda for the third year to include annotation tasks (named entity annotation and deep syntactic/semantic annotation) the feasibility of which we did not foresee in our original plan.

One task we performed was **updating an existing named entity recognition dataset** (the Szeged NER Corpus of business news) **to include a much richer entity annotation** than the original. Legacy Hungarian NER datasets and even the NerKor named entity corpus published that year only distinguished four entity types: person (PER), location (LOC), organization (ORG), and a category called 'miscellaneous' (MISC) covering all the rest. Words derived from names like *londoni* 'of London' and compounds containing names remained unannotated in these corpora. We enriched this

annotation turning it into one that covers 29+10 entity types instead of the original four. We distinguished subtypes of locations (facilities, geopolitical entities, and geographical locations) and of the 'MISC' category (events, feasts, media, products, projects, works of art, laws and norms, awards, securities, stock exchange indexes). We added annotation for dates, times, durations, and quantificational expressions (cardinal, ordinal, quantity, money, percentage). The annotation covers items derived from names or compounds thereof. We also added annotation for URL's, languages and for adjectives pertaining to nationality, religion, and political affiliation. We also annotated qualifiers for named entites (these are the +10 entity types annotated) when they were used in appositional constructions or otherwise modifying a named entity (like occupations, types of organizations etc.). We removed repetitive boilerplate-like content (8.5% of the original corpus), still, the new version **covers 2.8 times as many annotated spans as the original**.

We utilized **zero-shot cross-lingual transfer** to initialize the enrichment of entity types using three neural NER models: two of them are based on the English OntoNotes corpus, another one is based on the Czech Named Entity Corpus. The output of the models was automatically merged with the original NER annotation, and then automatically and manually corrected and further enriched. The most frequent error of the zero-shot models was including a definite article in the name spans (for organizations, works of art, etc.). We also generated a gazetteer from the lemmatized output of the models, and we created regex-based automatic correction patterns from inconsistencies and errors found in the gazetteer as well as other frequent error patterns observed in the preannotated corpus. The patterns also handled inflected forms. Final manual correction was performed using the collaborative INCEpTION annotation platform.

We evaluated the zero-shot performance of the original models, and we trained and evaluated a new model fine-tuning the Hungarian BERT model *huBERT* on the corpus. The best-performing zero-shot model finetuned by the FLAIR team on the English Ontonotes 5 corpus from XLM-RoBERTa achieved F1=0.752 on tags present both in Ontonotes and the final annotation and F1=0.675 on all tags, however, applying only one simple correction pattern removing definite article from name spans resulted improved this to F1=0.879/F1=0.806. This is what has made our annotation procedure very efficient. Our final model trained on the final corpus using the monolingual Hungarian language model huBERT still performs much better at F1=0.926.

The other zero-shot-transfer-based experiment we performed was **evaluating the performance of parser models trained on other languages (English and Czech) generating various meaning representations on Hungarian data**. The models we considered included a) Elementary Dependency Structures (EDS) based on English Resource Grammar with the underlying theory being HPSG/MRS (minimal recursion semantics), b) Discourse Representation Graphs (DRG) based on DRT (Discourse Representation Theory) and c) Prague Tectogrammatical Graphs (PTG) based on Prague Functional Generative Description (FGD). The EDS and DRG models were originally trained on English data, while two PTG models are available: one trained on the Prague Tectogrammatical Annotation in the Czech Prague Dependency Treebank (PDT), the other one on the English side of the Prague Czech–English Dependency Treebank (PCEDT).

Our initial probing of the models showed that the PTG parsers (especially the one trained on Czech data) have reasonable performance on Hungarian input, while the EDS and DRG parsers seemed to have much more serious problems interpreting Hungarian. We thus selected the much more promising **Czech and English PTG parsers for an in-depth evaluation** on a 150-sentence fragment of the Szeged Treebank manually correcting the output of the parser trained on Czech data and comparing this to the zero-shot annotations generated by the two PTG parser models. We needed to train ourselves during the process in PDT tectogrammatical annotation, reiterating and rediscussing

our solutions several times to converge on an annotation that we considered consistent with what is described in the PDT Tectogrammatical annotation guidelines.

Our findings concerning the performance of the models were:

1. Lemmata output by the models were mostly unusable because Czech or English lemmatization patterns applied to Hungarian data result in invalid output in most cases. This is a minor problem though, since there are good lemmatizers for Hungarian.
2. In contrast, grammatical/semantic relations among content words (edge labels in the graph, `functors' and `subfunctors' in PDT terminology) carry over relatively well to Hungarian (F1=0.7). The model is especially good at identifying adjuncts (time, place, directional and manner adverbials). Unfortunately, the annotation of predicate argument relations in PDT is in most cases is limited to two relations called *ACT* and *PAT.* These have nothing to do with real thematic roles: ACT is the subject, PAT is the second most prominent argument (and in the case of copula constructions, it is the predicate(!)).
3. Part-of-speech annotation is quite accurate with only some adverbs being mistagged as adjectives.
4. Annotation includes a feature on topic-focus articulation, which is an advanced feature. However, it only has three different values (t, f, c), while we think there should be four, and the annotation manual said practically nothing about this feature.
5. Unfortunately, many relevant grammatical features present in PDT were omitted when converting it to PTG on which the parser was trained (number, person, tense, modality, degree etc.).
6. In addition to edge labels, the model also relatively successfully predicted empty elements, such as dropped pronouns and ellipsis as long as similar patterns apply to both the source and the target language. The model trained on Czech was able to predict dropped subjects, control, quasi-control, other coreference relations, and existentially bound optional arguments. It was also able to handle gapping in the second clause in elliptic constructions. However, it cannot predict dropped objects or possessors or properly handle gapping in the first conjunct, as these constructions do not occur in Czech.

Overall, the model yielded reasonable performance, and we found that it could be feasibly applied in a semi-automatic annotation scenario. Transfer from Czech to Hungarian performed better than English to Hungarian because the source and the target language share more typological characteristics like rich morphology, free word order, pro drop etc. even though they belong to different language families.

In the first three years of the project, we also performed **semantic classification of various adverbial adjunct constructions** (adverbial participles, negative participles, adverbs of manner, and specific locative constructions), created an embedding-based model for identifying bogus compound analyses and invalid lemmatization/morphological annotation, **investigated coordinated structures and the semantics of compounds**. These lines of research were presented in the report on the PI's related PD125216 project.

In the fourth year of the project, we **performed an update** similar to the one performed in the previous year **to a recent and much bigger named entity recognition dataset (of over one million tokens), NYTK-NerKor** (Simon and Vadász, 2021), that, in contrast to the single-domain Szeged NER corpus, has a broad coverage of domains and topics: fiction, legal texts, web-crawled content including user-generated text, news, and a portion of the Hungarian Wikipedia.

We enriched the annotation turning it into one that covers 29 entity types instead of the original four. The new version thus contains 7 times as many distinct entity types and about twice as many annotated spans as the original version.

We applied our previously used methodology for the update utilizing zero-shot cross-lingual transfer using the same three neural NER models, the output of which was automatically merged with the original NER annotation, and then automatically and manually corrected and further enriched. The annotation features many types not present in the OntoNotes annotation scheme, e.g. time duration, age, media (journals, tv stations and news portals) and even some that were not present in the business news corpus annotated in the previous year, e.g. social media. These were introduced semi-automatically and corrected manually.

As an experiment on introducing a new entity subtype to the corpus, we also added a 12000-token subcorpus on cars and other motor vehicles (distinguishing motor vehicles as a subtype of the product category). For training, we selected articles from the archive of the hvg.hu news site using motor-vehicle-related keywords. Then, we chose sentences from this collection that contained car makes and models that were present in the menu structure of a car dealer's website. We pre-annotated this subcorpus using the Flair OntoNotes model. We then manually corrected the annotation and replaced product tags by car tags for car names. The new corpus version is thus called NerKor+Cars-OntoNotes++.

We evaluated the zero-shot performance of the original models, and we trained and evaluated a new model fine-tuning the Hungarian BERT model huBERT on the corpus. The best-performing zero-shot model finetuned by the FLAIR team on the English Ontonotes 5 corpus from XLM-RoBERTa achieved F1=0.82 on tags present both in Ontonotes and the final annotation and F1=0. 783 on all tags after applying only a simple correction pattern removing the definite article from name spans. Our huBERT-based model trained on the final corpus performs much better at F1=0.8957. We have also found when comparing the performance of our model with another identically parametrized model trained on the original NerKor annotation that the division of some entity classes (especially MISC) into several subclasses did not impact performance negatively: the model trained on the original NerKor achieved F1=0.91 on named entities, the one trained on the new version of the corpus achieved F1=0.92.

The performance of the models is slightly worse on this diverse corpus than the results we obtained on the single-domain business news corpus, but these models have a much more stable cross-domain performance, performing significantly better outside of their training domain.

We released the corpus, NerKor+Cars-OntoNotes++, on GitHub and the trained models on the HuggingFace hub.

We used the models trained to introduce named entity and numerical/time expression annotation to our web-crawled corpus. We trained new word embedding models on the entity annotated and lemmatized version of the corpus obtaining entity embeddings. This model can be used to identify names having a similar distribution to the name submitted to it as a query. The model contains the vector representations not only of full entity names but also of their parts. This makes queries also on parts of names such as surnames and first names possible. Using these models, **we created an anonymization/pseudonymization model prototype** that

1. identifies names and their type in the input text,
2. creates a lemmatized list of the entities identified,
3. generates a randomized list of distributionally similar replacement entities for each entity in the list

(this configuration can be manually modified), and
4. replaces entities in the text based on the configuration generated reinflecting the names to fit the context.

Substitutes for the names of persons are generated in such a way that the surname and the first name are replaced individually. This results in a consistent replacement of the surname of persons belonging to the same family keeping such relations consistent in the pseudonymized version of the text.

The model has some shortcomings that may be subject to later enhancements, such as nicknames are not replaced consistently with the corresponding formal name, the consistency of generated addresses is not checked (the zip code may not apply to the settlement, etc.) male names may be replaced by female names and vice versa, etc.

We also added many frequent names identified in the corpora to the morphological database of the Humor/emMorph morphological analyzer.

Originally, our project would have ended after the fourth year. However, the third and fourth years coincided with lockdowns due to the COVID pandemic. During the pandemic years, we could not participate in conferences in person, which resulted in considerable savings on conference costs. We could utilize these savings along with other unspent amounts reallocating them for a further year of research after our request to extend the project was accepted.

In the fifth extra year, we returned to the central theme of our first year: questions. However, we wanted to come up with something practical: a model that can answer questions based on some given text in Hungarian. Such models need not only be created, but it is also necessary to measure their performance. We thus **created a question answering benchmark dataset for Hungarian**. We largely **followed the principles of the English SQuAD 2.0 data set**, however, like in some more recent English question answering datasets, we introduced some innovations. SQuAD is an extractive QA dataset in the sense that answers are simply marked as single spans of words (as if we used a highlighter).

Similarly to SQuAD 2.0, the corpus is characterized by the following: a) high-quality Wikipedia articles serve as context for the questions, b) factual (not opinion-type) questions are included, c) also contains questions that are not answered in the given text, d) in the original text, we marked the shortest possible answer to the given question (if any), e) when formulating the questions, we paraphrased the original text, so in most cases the answer cannot be found using a lexical search, f) the questions can be interpreted not only in the context of the given text, but also as independent questions (e.g. they do not contain unanchored pronouns).

Compared to SQuAD, **we introduced** the following **innovations** (special question types are explicitly marked in the database): a) There may be more than one short answer to the given question in the given text (list type answer, approx. 8.5% of the answered questions). b) In addition to the short answer, we also gave a long answer, which includes all the relevant information necessary to answer the question (min. 1 clause, often several sentences). c) It contains yes-no questions (about 9%). Here, in addition to the long answer containing the essential circumstances, an explicit yes/no answer is also specified (or the lack of a clear binary answer is indicated). d) The unanswerable questions (about 28.3% of the questions) are relevant questions related to the given topic, not questions generated by substitution from questions having an answer. e) There are also questions that can only be answered after performing counting or arithmetic operations (similarly to the DROP database). Calculations involve counting of listed elements, calculation of dates, durations and other quantities

with simple arithmetic operations. f) Some of the unanswerable questions are tricky questions, where people would easily infer an answer from the text based on wrong default assumptions. These cases were marked separately, and the assumed answer was also indicated. g) If the original span in the text does not correspond to the form in which the given question should be answered (e.g. the original case ending is not appropriate), we have provided the form of the answer appropriate in the context of the question. This latter information can be used to train a generative model that can adapt the answer to be pragmatically adequate to the question. The database contains a total of 23,700 (17,000 answerable and 6,700 unanswerable) questions. Questions were created for 142 Wikipedia articles. The following table summarizes the number of types of questions in the dataset.

Type	number	ratio
There is an answer	**16992**	71.67%
. Yes-no	1621	9.20%
. . Yes	859	52.99%
. . No	638	39.36%
. . Uncertain	124	7.65%
. Not an extractive answer	4452	26.20%
. Arithmetics	427	2.51%
. List	1455	8.56%
. Not SQuAD-compatible	5203	30.62%
No answer	**6716**	28.33%
. Tricky no answer	629	9.37%
Sum	**23708**	100.00%

We implemented and evaluated a set of baseline retrieval and answer span extraction models on the dataset.[1] The latter type of models can be used to identify the answer in a given context. The former serve as a means to retrieve candidate documents relevant for the question. A simple BM25 model performed better than any vector-based solution for retrieval. This is not surprising, as we had access only to models that were not tuned to the question-document retrieval task or covered only English. Whet training reader models, we again found that cross-lingual transfer (from English) significantly improved performance: models pretrained on SQuAD 2.0 performed much better than a Hungarian model trained on our dataset only.

Resources and models related to this project can be found at https://users.itk.ppke.hu/~sikbo/fk125217.

[1] Due to the lack of adequate hardware resources, and to foster cooperation, we cooperated with researchers from Szeged University when creating and testing the models.