Automatic Animacy Classification for Latvian Nouns

Ralfs Brutāns
University of Amsterdam
Vrije Universiteit Amsterdam
ralfs.brutans@student.uva.nl

Jelke Bloem
University of Amsterdam
j.bloem@uva.nl

Abstract

We introduce the first automatic animacy classifier for the Latvian language. Animacy, a linguistic feature indicating whether a noun refers to a living entity, plays an important role in Latvian grammatical structures and syntactic agreement, but remains unexplored in Latvian NLP. We adapt and extend existing methods to develop type-based animacy classifiers that distinguish between human and non-human nouns. Due to the limited utility of Latvian WordNet, the classifier's training data was derived from the WordNets of Lithuanian, English, and Japanese. These lists were intersected and mapped to Latvian nouns from the Tēzaurs dictionary through automatic translation. The resulting dataset was used to train classifiers with fastText and LVBERT embeddings. Results show good performance from a MLP classifier using the last four layers of LVBERT, with Lithuanian data contributing more than English. This demonstrates a viable method for animacy classification in languages lacking robust lexical resources and shows potential for broader application in morphologically rich, under-resourced languages.

1 Introduction

There are many languages that are bound to go extinct despite efforts to preserve them, while others, such as English, are widely spoken and face no such risk. The Latvian language is positioned somewhere in between (Jansone, 2010), meaning that its long-term survival depends on active efforts to maintain and develop it. As the official language of Latvia and one of the official languages of the European Union, Latvian has around 1.5 million native speakers[1], significantly fewer than global languages like English. This smaller speaker base also means that Latvian is considerably less researched in fields such as natural language processing (NLP) (Laucis and Jēkabsons, 2021). Ensuring that Latvian keeps pace with advances in NLP is essential not only for preserving and modernizing the language but also for supporting its use in digital applications such as machine translation and automated text processing.

One important but underexplored linguistic feature in NLP research, particularly for morphologically rich languages like Latvian, is animacy. Animacy refers to how "alive" or independently acting a noun's referent is — humans and animals are animate, while objects are not. This distinction is encoded in the grammars of many natural languages, influencing word order, case marking, and agreement patterns. Studies suggest that incorporating animacy into computational models can enhance machine translation and parsing accuracy (Øvrelid, 2006, 2008), in addition to informing linguistic studies. However, for Latvian, we are not aware of the existence of any animacy classifier.

We present an approach to animacy classification for Latvian nouns based on static and contextual word embeddings. While other animacy classifiers for under-resourced languages have used this approach (Tepei and Bloem, 2024 for Romanian), our approach is novel in relying on lexical-semantic resources for higher-resource languages. Previous approaches rely on WordNet hypernym relations to obtain a seed set of animacy-labeled data for supervised learning, but as the Latvian WordNet does not yet have a full tree of hypernym and hyponym relations, we instead rely on the WordNets for higher-resource languages and automatic translation to obtain such data.

2 Related Work

Three types of animacy are usually distinguished: grammatical, biological and conceptual (de Swart

[1] https://valoda.lv/valsts-valoda/

Proceedings of the Workshop on Beyond English: NLP for all Languages in an Era of LLM from Texts associated with RANLP 2025, pages 90–97, Varna, Bulgaria, Sep 12, 2025.
https://doi.org/10.26615/978-954-452-105-9-011

and de Hoop, 2018). Entities that possess physical characteristics such as the ability to die are said to as biologically animate. The speaker's perspective and cultural upbringing serve as the foundation for conceptual animacy. The way that people personify or give non-living things agency reflects this, for example, in mythology. Grammatical animacy, however, illustrates how a language's grammar reflects both biological and conceptual animacy. It functions as a condition or semantic feature that affects linguistic structures such as case marking and verb agreement. Usually, the concept of animacy is seen as being a continuous scale ranging from humans to inanimate abstract objects. According to Yamamoto (2006, p. 36), animacy is a matter of gradience determined by the overall animacy scale, hierarchy of persons, the agency scale, and the individuation scale.

Despite animacy usually being viewed as existing on a continuous scale or a hierarchy (DeLancey, 1981), natural language requires that concepts be categorized. The two most simple categorizations would be distinguishing between animate and inanimate nouns or distinguishing between human and non-human nouns. Most effects of grammatical animacy are based on a binary split, with tripartite systems being rare. As a consequence, most NLP literature on animacy also discusses the distinction between two to three categories.

Latvian does not mark animacy distinctions directly, but animacy has several effects on the grammaticality and felicitousness of sentences. Latvian has a somewhat free word order; however, there are more and less common sentence structures. Even though all six word order variations for subject-verb-object placement are possible, the most common structure is subject-verb-object (SVO) when the subject is animate, and object-verb-subject (OVS) is more frequent with inanimate subjects (Voits, 2014). Seržant and Taperte (2016) found that animacy plays a role in influencing the choice between accusative and nominative case marking in the Latvian debitive construction. Animate NPs also appear to be more likely to trigger genitive agreement, where the predicate agrees with the noun in the genitive case rather than with the quantifier. Inanimate NPs, on the other hand, do not favor the genitive or quantifier agreement (Kalnača and Lokmane, 2022, p. 85). These interactions between animacy and the felicitousness and grammaticality of sentences in Latvian suggest that animacy could be a useful feature also for downstream NLP tasks for Latvian, such as coreference resolution.

Animacy classifiers have been made for other languages, including under-resourced ones. Some of the first research on automatic animacy classification for nouns was done by Øvrelid (2004) on animacy classifiers for Norwegian (Øvrelid, 2005) and for Swedish (Øvrelid, 2008; Øvrelid, 2009). These classifiers are based on morphosyntactic features that were selected on linguistic grounds to classify into binary animate/inanimate categories. These classifiers achieved quite good results, achieving up to a 98.6% accuracy on unseen nouns. However, they are based on large pre-annotated animacy corpora, which is not something available for many under-resourced languages.

Subsequently, Bowman and Chopra (2012) proposed a classifier that classifies nouns into ten categories. This paper highlights the problem of trying to classify nouns into more categories than are expressed in grammar. It is more difficult to discriminate animacy when grammaticality and felicitousness is only governed by human/non-human or animate/inanimate categories as models can then only rely on semantic cues, not syntactic or morphological ones. Bloem and Bouma (2013) present an animacy classification tool for Dutch, which combines type-based classification using distributional features with a seed set of noun types that were given an animacy label based on the Cornetto lexical-semantic database (Vossen, 2006). They tested classification with a two-way distinction (human/nonhuman) and a three-way distinction (animate human/animate nonhuman/inanimate). Classification for the human/nonhuman distinction, which corresponds to distinctions made in Dutch grammar, performed much better. Their best-performing classification algorithm was a K-nearest neighbour classifier.

More recent approaches have turned to transfer learning to overcome data scarcity and enhance generalization across tasks and languages. Transfer learning has been widely used in natural language processing to address the challenges posed by limited labeled data, especially in under-resourced languages. The approach involves reusing representations learned from a general task, such as language modeling, for more specific tasks like animacy classification. Pretrained models such as FastText (Bojanowski et al., 2017) and contextual models like BERT (Devlin et al., 2019) are commonly used. For Latvian, LVBERT (Znotiņš and Barzdiņš, 2020) provides pretrained embeddings that can be applied to downstream tasks with minimal task-specific training. Prior work suggests that transfer learning can support tasks requiring semantic generalization by utilizing knowledge encoded during pretraining (Ruder et al., 2019; Conneau et al., 2020). In the case of animacy, this includes properties such as agency and sentience, which may be implicitly captured by language models.

3 Methodology

3.1 Use of WordNets for animacy-annotated lists

Based on the results of previous work and because Latvian does not have a strict grammatical animacy distinction, the animacy classifier we make distinguishes between human and non-human classes, employing lemmas from a word list to make a type-based classifier. Inspired by the recent animacy classification work on the under-resourced Romanian by Tepei and Bloem (2024), we used hyponymy relations to extract a seed set of nouns that were labeled for animacy using WordNet.

This labeling stems from the hierarchical structure of WordNet: It consists of a hierarchy of synsets that are in hyponym and hypernym relations with each other, with specific synsets at the bottom and general synsets at the top, representing concepts such as *entity* and *event*. Under *entity*, we find concepts such as *life form* and *object*. By taking words from all synsets that are hyponyms of one such high-level synset, such as the one corresponding to human words, it is possible to obtain a large number of words that refer to human entities.

However, Latvian WordNet (Paikens et al., 2023) is still under construction and is largely lacking in hyponymy and hypernymy relations, not containing a full tree of synsets. Therefore, we developed a different approach based on higher-resource WordNets. More specifically, WordNets for three languages— Lithuanian (Garabík and Pileckyte, 2013), English (Fellbaum, 1998) and Japanese (Bond and Kuribayashi, 2023) were used to extract lists of human and non-human nouns. The English WordNet was used for its interpretability, and Japanese WordNet was used for its large word tree containing a vast amount of noun relations and for the fact that it is not an Indo-European language, typologically distinct from both Latvian and English. In contrast, the Lithuanian WordNet was used because of its similarity with Latvian, the only other living Baltic language.

As in previous work, we constructed the seed sets of words with animacy labels by identifying high-order hypernyms that contained no instance of the other class for the three languages, namely, *asmuo, person* and 人 (all with the meaning person) as the human targets for Lithuanian, English, and Japanese, respectively. For the non-human class, all other unique beginner synsets that do not contain person as a hyponym were used for the inanimate class. After having established these high-order hypernym synsets, lists of all their hyponyms were extracted to obtain human and non-human nouns for all three languages.

3.2 Translating Latvian nouns

We used the online lexical resource Tēzaurs (Spektors et al., 2025) to obtain a dataset of Latvian nouns. All of these unique noun lemmas were automatically translated to Lithuanian, English, and Japanese for comparison with the extracted lists of animacy-labeled lists of nouns. To translate these nouns, we used the Google Translate API[2], which was found by Rikters (2015) to perform well for English-Latvian translation at the time.

The translations of the nouns were checked against the animacy-labeled lists of nouns in the three languages, and if a word was present in all three animacy-labeled sets with the same label, then its Latvian counterpart was included in an animacy-labeled list for Latvian with the corresponding label of human or non-human. This restriction reduces the possibility that translation errors between particular language pairs affect the quality of our seed set, as there was no manual translation quality control and polysemous words could have been translated incorrectly. In the end, the list consists of 5183 nouns, of which 735 are labeled as human.

3.3 FastText-based classifiers

We used pre-trained fastText embeddings (Bojanowski et al., 2017) to obtain static word vectors for Latvian nouns. FastText was chosen for its subword modeling capabilities, making it effective for morphologically rich languages, and prior work found it to be the best-performing static embedding for Latvian (Laucis and Jēkabsons, 2021).

[2]The API was called using the deep-translator library for Python: `https://pypi.org/project/deep-translator/`

Vectors were 300-dimensional, with character n-grams of length 5, a window size of 5, and 10 negative samples, trained on Common Crawl and Wikipedia. Each vector was paired with the animacy label derived from the WordNet intersection. We trained classifiers using K-nearest neighbors (KNN) ($k = 5$), Random Forest (RF) (100 estimators, Gini criterion), and Multi-Layer Perceptron (MLP) (hidden size 100, $\alpha = 0.0001$, learning rate 0.001) algorithms. We used these classifiers because they were used by Tepei and Bloem (2024), who included them based on good performance in previous work.

3.4 LVBERT-based classifiers

To explore the potential of contextual embeddings, we used LVBERT (Znotiņš and Barzdiņš, 2020), a transformer model trained on Latvian corpora. For each noun token, we extracted either layer 0 (non-contextualized) or a 3072-dimensional vector from the concatenation of the final four layers (layers 9-12, following Hosseini et al. 2023).

We try the latter approach because deeper layers capture richer semantic information (Devlin et al., 2019), and concatenation has shown strong performance in semantic similarity tasks. The first non-special token (i.e., the noun) was used for classification. Layer 0 embeddings were also tested to compare performance with fastText, as lower transformer layers may behave like static embeddings (Vulić et al., 2021). We cannot tune LVBERT for the task with a token classifier head as no corpus with animacy annotation in context is available.

3.5 Evaluation methods

To assess the quality of the static and contextual embedding-based classifiers, we split the labeled noun types into an 80%/20% train/test set, which was then used to perform a type-based evaluation of the classifiers.

We also perform a token-based evaluation because it represents a more naturalistic use setting despite the classifier being type-based. To this end, we chose, compiled, and cleaned nine random Wikipedia articles. Using the Python library Stanza (Qi et al., 2020), which has a POS tagger for Latvian trained on the Universal Dependencies treebanks for Latvian (Pretkalniņa et al., 2018), a list of nouns present in the texts was extracted and manually annotated for animacy by a native Latvian speaker. As the classifier is type-based, the nouns are lemmatized before annotation and prediction. Lemmas representing human collectives (e.g., *valdība* 'government') were assigned the non-human category due to them being treated as inanimate in Latvian grammar, exemplified by the use of demonstrative pronouns instead of personal pronouns. The classifiers were then used to predict class membership for the given nouns. Although the classifier is type-based and does not consider the surrounding context of nouns, token-based evaluation can provide a better benchmark of the classifier's performance in a naturalistic setting.

4 Results

4.1 Type-based evaluation

4.1.1 Results for the classifiers made with fastText embeddings

Classifier	Acc.	Pre.	Rec.	F1
KNN	0.857	0.780	0.222	0.345
RF	0.878	**0.915**	0.307	0.461
MLP	**0.900**	0.728	**0.653**	**0.688**

Table 1: Type-based evaluation performance of fastText-based classifiers. Baseline accuracy is 0.830.

For fastText-based classification of noun types, the RF algorithm achieves a higher precision score of 91.5% against the KNN and MLP models (see table 1). However, the MLP algorithm shows better recall and accuracy scores of 65.3% and 90.0%, respectively. This entails that when the RF model predicts the human class, it is almost always correct (precision); however, it is very conservative in labeling nouns as human, leading to very low recall of 30.7% (false negatives). On the other hand, the high accuracy and recall scores for MLP show that it is overall quite good at predicting class membership, and it achieves the highest F1 score of 68.8%. KNN shows the worst performance overall. Baseline accuracy for this dataset is 83%, as 83% of the test nouns are non-human.

4.1.2 Results for classifiers trained with LVBERT embeddings

Using layer 0 embeddings from LVBERT (Znotiņš and Barzdiņš, 2020) for classifier training did not prove to be useful (see table 2), yielding worse scores than their fastText counterparts (with the exception of KNN).

Results with the last four layers of LVBERT embeddings are better, with the MLP classifier clearly outperforming the KNN and RF algorithms.

Classifier	Acc.	Pre.	Rec.	F1
KNN	0.854	0.602	0.420	0.494
RF	0.842	0.620	0.176	0.274
MLP	**0.880**	**0.697**	**0.523**	**0.597**

Table 2: Type-based evaluation on LVBERT layer 0-based classifiers. Baseline accuracy is 0.830.

Classifier	Acc.	Pre.	Rec.	F1
KNN	0.905	0.824	0.557	0.664
RF	0.849	**0.913**	0.119	0.210
MLP	**0.916**	0.795	**0.682**	**0.734**

Table 3: Type-based evaluation performance of LVBERT last four layer concatenation-based classifiers

Classifier	Acc.	Pre.	Rec.	F1
KNN	0.905	0.717	0.349	0.469
RF	**0.931**	**0.911**	0.468	0.618
MLP	0.894	0.542	**0.771**	**0.636**

Table 4: Token-based evaluation on fastText-based classifiers. Baseline accuracy is 0.880.

Classifier	Acc.	Pre.	Rec.	F1
KNN	0.910	0.636	0.578	0.606
RF	0.896	**1.000**	0.138	0.242
MLP	**0.938**	0.768	**0.697**	**0.731**

Table 5: Token-based evaluation on LVBERT-based classifiers

As shown in Table 3, the MLP classifier achieves the highest accuracy (0.916), recall (0.682), and F1 score (0.734) among the three LVBERT-based classifiers, along with a strong precision score (0.795). This indicates that the LVBERT-based MLP classifier is both relatively accurate and balanced in predicting the "human" class. The RF classifier, while achieving the highest precision (0.913) among the three, performs poorly in recall (0.119) and F1 score (0.210). This reflects a conservative approach in labeling tokens as "human," resulting in many false negatives. The KNN classifier surpasses the fastText-based KNN model. The LVBERT-based MLP classifier outperforms the fastText-based MLP classifier. Only the LVBERT-based RF classifier does not outperform its fastText-based counterpart. The performance dynamics of the different algorithms remain the same in a type-based evaluation, where the RF algorithm has the highest precision, while MLP outperforms on the other three metrics. This suggests that richer, contextualized representations from transformer models are beneficial when classifying noun animacy at the type level.

4.2 Token-based evaluation

For the token-based evaluation with LVBERT, we only used the last four layer approach due to superior performance in the type-based evaluation. This evaluation aims to show whether a more naturalistic setting would affect the performance rankings of the classifiers. Nine random Wikipedia articles on different topics were chosen preprocessed. Next, a Latvian POS tagger trained on the UD (universal dependencies) treebank corpus for Latvian (Pretkalniņa et al., 2018) was employed using the Stanza (Qi et al., 2020) library for Python to obtain a list of 1342 noun lemmas for animacy labeling. All the lemmas were manually annotated by a native Latvian speaker with human/non-human labels based on the meaning of the word token in context. 46 lemmas were excluded from this test set due to POS-tagging errors or faulty text to obtain 1296 annotated lemmas, of which 908 were used for unseen prediction. Out of these 908 lemmas, 799 were annotated with the non-human label and 108 with the human label, setting the majority baseline accuracy at 88.0%

In this evaluation, the LVBERT-based classifiers generally outperform the fastText-based classifiers. The LVBERT-based MLP classifier achieved the highest accuracy (0.938) and F1 score (0.731) across all settings. It also had the highest recall (0.697), indicating stronger performance in identifying human-referent nouns. The LVBERT-based RF classifier, while achieving perfect precision (1.000), showed a very low recall (0.138), suggesting a highly conservative classification strategy that avoids false positives but misses many actual human nouns. For fastText, overall results are lower but still competitive, and RF classifiers perform better here than they do on LVBERT embeddings.

4.3 Language resource ablation

As our methodology involves combining data from higher-resource WordNets, we also evaluate the contribution of each source language WordNet by training and testing classifiers using only one of the languages as source data. We perform the token-based evaluation for all nouns that are not in the training data (unseen nouns). This does mean that each classifier has a different test set, as some lan-

Language	Acc.	Pre.	Rec.	F1
Lithuanian	**0.932**	**0.680**	**0.742**	**0.710**
English	0.862	0.383	0.561	0.455
Japanese	0.908	0.632	0.655	0.643

Table 6: Token-based evaluation on LVBERT-based MLP classifiers trained on single language WordNets. Baseline accuracy differs per language.

guages have labels for more nouns than others, so the results are not directly comparable, but it does give an impression of the relative contribution of each resource. Specifically, for the English-based classifier there are 398 unseen nouns in our evaluation set, for Japanese there are 434 and for Lithuanian (the smallest WordNet) there are 790. For comparison, the original token-based evaluation had 908 unseen nouns (not occurring in all three resources). We perform this experiment in the best-performing setting, using the last four layers of LVBERT with the MLP classifier. The results are shown in Table 6. We observe that the classifier based on Lithuanian WordNet outperforms the others, despite this resource being the smallest (6357 noun synsets, compared to 82,115 for English and 42,737 for Japanese). Latvian and Lithuanian are closely related typologically, with both being East Baltic languages. This result suggests that typological relatedness is more beneficial than resource comprehensiveness for transfer learning for animacy classification in a natural language setting. However, the approach of combining all three language resources still outperforms Lithuanian only (0.731 vs 0.710 F1 score).

5 Discussion

This study introduced the first classifiers for predicting animacy (human vs. non-human) in Latvian nouns, using a methodology adapted for a low-resource setting. We evaluated 12 classifiers based on fastText and LVBERT embeddings, with animacy-labeled training data derived through multilingual WordNet intersection and translation. While we found that training data from a typologically related language was more useful, the best results were achieved by LVBERT-based MLP classifiers using the final four layers of the model trained on labels from an intersection of three languages' WordNets. These outperformed fastText-based models in both type- and token-based evaluations, with the best model achieving 93.8% accuracy on unseen nouns.

Although all classifiers were trained on type-level data, token-based evaluation showed that contextual embeddings can generalize well to more naturalistic usage, even without explicit token-level supervision. Layer 0 LVBERT embeddings, which behave more like static vectors, underperformed compared to deeper contextual layers. The success of LVBERT shows that transformer-based representations can be beneficial even in the absence of large annotated corpora.

Another promising direction is to use generative large language models' zero-shot generalization capability. Recent work demonstrates that GPT-3 can distinguish animate/inanimate entities in zero-shot settings across languages (Pucci et al., 2025), though this has not yet been explored in a classification task. Probing or fine-tuning LLMs such as LVBERT, LitLat BERT, or multilingual open-weight models (e.g., Gemma, LLaMA) on animacy tasks could offer new insights and performance improvements. Evaluating how well such models generalize animacy features to under-resourced languages would help clarify their linguistic competence and applicability in downstream NLP tasks.

6 Conclusion

We present the first type-based approach to animacy classification for Latvian nouns using cross-lingual projection and multilingual lexical resources. Animacy-labeled word lists were automatically constructed by aligning English, Lithuanian, and Japanese WordNets with Latvian nouns from the Tēzaurs dictionary via translation. This enabled training data creation without manual annotation. We trained classifiers using fastText and contextual LVBERT embeddings. Results showed that LVBERT-based models—especially MLP with concatenated final layers—outperformed fastText models in both type- and token-based evaluations. While RF classifiers achieved the highest precision, MLPs offered better balance overall. A language ablation study showed the most typologically related language to contribute more.

This work demonstrates the feasibility of animacy classification in low-resource languages without native WordNets. Despite limitations—such as label noise from translation and lack of context in static embeddings—our approach lays a foundation for extending animacy annotation and classification to other languages.

7 Limitations

Several limitations remain. Training data labels were derived via automatic translation, which may introduce noise. Furthermore, type-based classifiers cannot resolve context-sensitive cases of animacy, such as polysemous words (*medijs*: psychic or media in Latvian). In a naturalistic setting, our classifier would have to be used after lemmatization, and imperfect lemmatization due to the extensively inflected nature of Latvian might reduce accuracy. Future work could focus on building token-level classifiers, such as by tuning LVBERT. This would require a corpus where nouns are annotated for animacy in context, which is currently unavailable for Latvian. Another direction would be to address the class imbalance in training data by augmenting the human noun class through synonym expansion.

8 Ethical considerations

We do not foresee any particular harmful impacts of this work. While the pre-trained embeddings we use may encode harmful biases, we could not identify any reason to assume that these biases pertain to the human/nonhuman distinction that we classify. Most concerns regarding bias identified in the literature pertain to social identities that differ between humans (e.g. gender bias). When deploying animacy classification of the type we propose, we do recommend to evaluate that people with protected characteristics relevant to the use case aren't more likely to be misclassified as nonhuman, as this may cause harm.

References

Jelke Bloem and Gosse Bouma. 2013. Automatic animacy classification for Dutch. 3:82–102.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *arXiv:1607.04606 [cs]*.

Francis Bond and Takayuki Kuribayashi. 2023. The Japanese WordNet 2.0. *ACL Anthology*, pages 179–186.

Samuel R. Bowman and Harshit Chopra. 2012. Automatic Animacy classification. In *Proceedings of the NAACL HLT 2012 Student Research Workshop*, pages 7–10, Montréal, Canada. Association for Computational Linguistics.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Scott DeLancey. 1981. An interpretation of split ergativity and related patterns. *Language*, 57:626–657.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies*, volume 1, pages 4171–4186.

Christiane Fellbaum. 1998. *WordNet : an electronic lexical database*. Mit Press.

Radovan Garabík and Indre Pileckyte. 2013. From multilingual dictionary to Lithuanian WordNet.

MohammadSaleh Hosseini, Munawara Munia, and Latifur Khan. 2023. BERT has more to offer: BERT layers combination yields better sentence embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 15419–15431, Singapore. Association for Computational Linguistics.

Ilga Jansone. 2010. Latvian. *Revue belge de philologie et d'histoire*, 88(3):741–764.

Andra Kalnača and Ilze Lokmane. 2022. Partitive genitive constructions and agreement variations in Latvian. *Linguistic Variation*.

Rolands Laucis and Gints Jēkabsons. 2021. Evaluation of word embedding models in Latvian NLP tasks based on publicly available corpora. *Applied Computer Systems*, 26:132–138.

Lilja Øvrelid. 2005. Animacy classification based on morphosyntactic corpus frequencies: some experiments with Norwegian nouns. In *Proceedings of the Workshop on Exploring Syntactically Annotated Corpora*, pages 24–34.

Lilja Øvrelid. 2009. Empirical evaluations of Animacy annotation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 630–638, Athens, Greece. Association for Computational Linguistics.

Peteris Paikens, Agute Klints, Ilze Lokmane, Lauma Pretkalniņa, Laura Rituma, Madara Stāde, and Laine Strankale. 2023. Latvian WordNet. *ACL Anthology*, pages 187–196.

Lauma Pretkalniņa, Laura Rituma, and Baiba Saulīte. 2018. Deriving enhanced universal dependencies from a hybrid dependency-constituency treebank. *Lecture notes in computer science*, pages 95–105.

Giulia Pucci, Fabio Massimo Zanzotto, and Leonardo Ranaldi. 2025. Animate, or inanimate, that is the question for large language models. *Information*, 16(6):493.

Peng Qi, Yuhao Zhang, Yuhui Zhang, Jason Bolton, and Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. Association for Computational Linguistics.

Matīss Rikters. 2015. Multi-system machine translation using online apis for English-Latvian.

Sebastian Ruder, Matthew E. Peters, Swabha Swayamdipta, and Thomas Wolf. 2019. Transfer learning in natural language processing. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 15–18, Minneapolis, Minnesota. Association for Computational Linguistics.

Ilja A Seržant and Jana Taperte. 2016. Differential argument marking with the Latvian debitive. pages 199–258.

Andrejs Spektors, Lauma Pretkalniņa, Normunds Grūzītis, Pēteris Paikens, Laura Rituma, Baiba Saulīte, Gunta Nešpore-Bērzkalne, Ilze Lokmane, Agute Klints, Madara Stāde, Mikus Grasmanis, Ilze Auziņa, Artūrs Znotiņš, Roberts Darģis, and Guntis Bārzdiņš. 2025. Tēzaurs.lv 2025 (spring edition). *Clarin.lv*.

Peter de Swart and Helen de Hoop. 2018. Shifting animacy. *Theoretical Linguistics*, 44:1–23.

Maria Tepei and Jelke Bloem. 2024. Automatic animacy classification for Romanian nouns. *ACL Anthology*, pages 1825–1831.

Toms Voits. 2014. Discourse-related word order variation in Latvian. *Valoda: nozīme un forma*, pages 144–156.

PTJM Vossen. 2006. Cornetto: Een lexicaal-semantische database voor taaltechnologie. *Dixit*, (special issue).

Ivan Vulić, Edoardo Maria Ponti, Anna Korhonen, and Goran Glavaš. 2021. LexFit: Lexical fine-tuning of pretrained language models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5269–5283, Online. Association for Computational Linguistics.

Mutsumi Yamamoto. 2006. *Agency and impersonality : their linguistic and cultural manifestations*. J. Benjamins Pub. Co.

Artūrs Znotiņš and Guntis Barzdiņš. 2020. LVBERT: Transformer-based model for Latvian language understanding. *Frontiers in artificial intelligence and applications*.

Lilja Øvrelid. 2004. Disambiguation of syntactic functions in norwegian: Modeling variation in word order interpretations conditioned by animacy and definiteness. *Proceedings of the 20th Scandinavian Conference of Linguistics.*

Lilja Øvrelid. 2006. Towards robust animacy classification using morphosyntactic distributional features. *ACL Anthology*, Student Research Workshop:47–54.

Lilja Øvrelid. 2008. Linguistic features in data-driven dependency parsing. *ACL Anthology*, pages 25–32.