Pyramidal Recursive Composition of Multi-Word Units into Unified Representations

Piramidalna rekurzivna kompozicija višerječnih izraza u objedinjene reprezentacije

Karlo Babić, Ana Meštrović
University of Rijeka
Faculty of Informatics and Digital Technologies
Radmile Matejčić 2, Rijeka, Croatia
{karlo.babic, amestrovic}@uniri.hr

Abstract. *In this paper, we explore the composition of word embeddings to create richer, more meaningful representations of multi-word units. Existing methods, such as averaging word embeddings, provide simple and efficient approaches. However, they often fail to capture the complexity of multi-word interactions. To address this, we employ the Pyramidal Recursive learning (PyRv) method, which recursively combines word embeddings into unified representations. Originally developed for constructing representations hierarchically from subwords to phrases, PyRv is well-suited for progressively merging individual word vectors into phrase vectors. We evaluate the effectiveness of PyRv for embedding composition using fastText embeddings on the dependency relation labeling task. Using a single fastText word embedding yields an accuracy of 71%. Averaging five fastText word embeddings (the middle word and its four neighboring words) results in a significant drop in accuracy to 34%. In contrast, by composing five word embeddings with PyRv, we achieve an accuracy of 77%, demonstrating the superior ability of PyRv to integrate multiple word embeddings into more expressive representations. These findings highlight the potential of PyRv as a lightweight yet powerful technique for word embedding composition.*

Keywords. Composition, embeddings, pyramidal, recursive neural network, text representation learning

Sažetak. *U ovom radu istražujemo kompoziciju vektorskih reprezentacija riječi (eng. "word embeddings") s ciljem stvaranja semantički bogatijih i smislenijih reprezentacija višerječnih izraza. Postojeće metode, poput prosječne vrijednosti vektorskih reprezentacija riječi, nude jednostavan i učinkovit pristup. Međutim, takvi pristupi često ne uspijevaju obuhvatiti složenost odnosa između riječi. Kako bismo to prevladali, koristimo metodu Pyramidal Recursive learning (PyRv), koja rekurzivno kombinira vektorske reprezentacije riječi u objedinjene reprezentacije. Izvorno razvijena za hijerarhijsku izgradnju reprezentacija od podriječi do fraza, PyRv je prikladna za postupno spajanje pojedinačnih vektora riječi u vektore fraza. Evaluaciju metode PyRv za kompoziciju vektorskih reprezentacija provodimo korištenjem fastText vektora riječi u zadatku označavanja zavisnosnih odnosa. Korištenjem fastText vektorske reprezentacije jedne riječi postiže se točnost od 71%. Prosječna vrijednost pet fastText vektora riječi (središnje riječi i njezine četiri susjedne) rezultira značajnim padom točnosti na 34%. Suprotno tome, kompozicijom pet vektora riječi metodom PyRv postiže se točnost od 77%, što pokazuje nadmoćnu sposobnost metode PyRv u integraciji više vektora riječi u izražajnije reprezentacije. Ovi rezultati ukazuju na potencijal metode PyRv kao lagane, ali snažne tehnike za kompoziciju vektorskih reprezentacija riječi.*

Ključne riječi. Kompozicija, vektorske reprezentacije, piramidalno, rekurzivna neuronska mreža, učenje reprezentacija teksta

1 Introduction

Word embeddings are foundational to many natural language processing (NLP) tasks, providing a way to map words into continuous vector spaces that capture semantic relationships between them. By converting words

1 Uvod

Vektorske reprezentacije riječi temelj su mnogih zadataka obrade prirodnog jezika (NLP), jer omogućuju preslikavanje riječi u kontinuirane vektorske prostore koji zadržavaju semantičke odnose među njima. Pre-

into numerical representations, word embeddings allow machines to process and understand text in a way that retains important linguistic properties. Popular word embedding methods, such as Word2Vec (Mikolov et al., 2013), GloVe (Pennington et al., 2014), and FastText (Bojanowski et al., 2017), have demonstrated the utility of these representations by positioning semantically similar words closer in the vector space. The effectiveness of NLP models often hinges on the quality of these embeddings, as richer and more informative representations can lead to improved performance across various downstream tasks.

In many real-world applications, however, understanding text at the word level alone is insufficient. The need to represent larger linguistic units, such as phrases or sentences, necessitates techniques for combining word embeddings into more complex structures. Word composition, which involves aggregating individual word embeddings to represent multi-word expressions or entire sentences, serves this purpose. By integrating information from multiple word vectors, compositional methods aim to capture both the meanings of individual words and the syntactic and semantic interactions between them, particularly in morphologically complex languages, such as Croatian, which was used for evaluation in this paper.

There are several established methods for combining word embeddings. Simple techniques include element-wise operations such as addition, averaging, or multiplication, which produce a composite vector by leveraging individual word vectors (Arora et al., 2017; Joulin et al., 2016). These methods, while computationally efficient, may fail to fully capture the complexity of phrase or sentence meaning. More sophisticated approaches employ weighted combinations, context-aware methods, or syntactic structures to improve the expressiveness of the resultant embeddings (Babić et al., 2020; Bahdanau, 2014; Socher et al., 2013).

While transformer-based models, such as BERT (Devlin et al., 2018) and GPT (Brown et al., 2020), offer robust pre-trained sentence embeddings by learning deep contextual representations, they are often computationally expensive and may not always align with the specific needs of certain tasks. Word composition methods provide a more lightweight and flexible alternative, especially in cases where transparency and control over the aggregation process are crucial. Additionally, word-level composition techniques can better retain the granularity of individual word meanings, which is sometimes diluted in sentence-level embeddings produced by transformer models.

Word composition provides a valuable approach for constructing meaningful representations of multi-word units, balancing computational efficiency and interpretability. These methods remain relevant, particularly in domains where sentence embedding techniques may obscure important details or where domain-

tvaranjem riječi u numeričke reprezentacije, vektori riječi omogućuju računalima obradu i razumijevanje teksta na način koji očuva važne jezične značajke. Popularne metode za generiranje vektorskih reprezentacija riječi, poput Word2Vec (Mikolov i dr., 2013), GloVe (Pennington i dr., 2014) i FastText (Bojanowski i dr., 2017), pokazale su korisnost ovih reprezentacija tako što pozicioniraju semantički slične riječi bliže jedna drugoj u vektorskom prostoru. Učinkovitost NLP modela često uvelike ovisi o kvaliteti ovih reprezentacija, budući da bogatije i informativnije reprezentacije mogu dovesti do boljih rezultata u različitim zadacima.

Međutim, u mnogim stvarnim primjenama, razumijevanje teksta na razini pojedinačnih riječi nije dovoljno. Potreba za reprezentacijom većih jezičnih jedinica, poput fraza ili rečenica, zahtijeva tehnike za kombiniranje vektora riječi u složenije strukture. Kompozicija riječi, koja podrazumijeva agregaciju pojedinačnih vektorskih reprezentacija kako bi se prikazale višerječne izraze ili čitave rečenice, ima upravo tu svrhu. Integriranjem informacija iz više vektora riječi, kompozicijske metode nastoje obuhvatiti značenja pojedinih riječi, kao i sintaktičke i semantičke odnose među njima – osobito u morfološki složenim jezicima, poput hrvatskog, koji je korišten u evaluaciji ovog rada.

Postoji više poznatih metoda za kombiniranje vektora riječi. Jednostavne tehnike uključuju elementne operacije, poput zbrajanja, prosječavanja ili množenja, koje stvaraju kompozitni vektor korištenjem pojedinačnih vektora riječi (Arora i dr., 2017; Joulin i dr., 2016). Iako su te metode računski učinkovite, često ne uspijevaju u potpunosti obuhvatiti značenje izraza ili rečenice. Složeniji pristupi koriste ponderirane kombinacije, metode osjetljive na kontekst ili sintaktičke strukture kako bi poboljšali izražajnost dobivenih reprezentacija (Babić i dr., 2020; Bahdanau, 2014; Socher i dr., 2013).

Iako modeli temeljeni na transformerima, poput BERT-a (Devlin i dr., 2018) i GPT-a (Brown i dr., 2020), nude snažne unaprijed naučene vektorske reprezentacije rečenica učenjem dubokih kontekstualnih značenja, često su računski zahtjevni i ne odgovaraju uvijek specifičnim potrebama određenih zadataka. Metode kompozicije riječi predstavljaju lakšu i fleksibilniju alternativu, osobito u slučajevima gdje su transparentnost i kontrola nad procesom agregacije od presudne važnosti. Osim toga, metode temeljene na kompoziciji na razini riječi bolje zadržavaju granularnost značenja pojedinih riječi, što se ponekad gubi u rečeničnim reprezentacijama koje generiraju modeli transformera.

Kompozicija riječi pruža vrijedan pristup za izgradnju smislenih reprezentacija višerječnih izraza, uravnotežujući računalnu učinkovitost i interpretabilnost. Ove metode ostaju relevantne osobito u područjima gdje tehnike vektorskog reprezentiranja rečenica mogu prikriti važne pojedinosti ili gdje je potrebna domensko-specifična prilagodba kompozicije vektora.

specific customization of embedding composition is required.

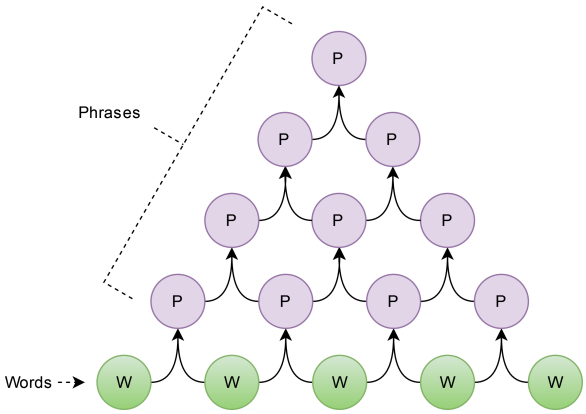
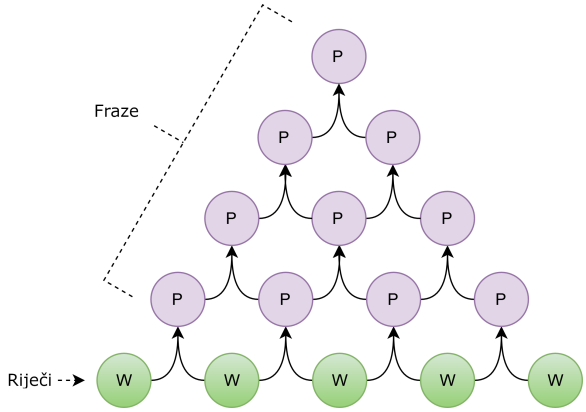


Figure 2: A visual representation of pyramidal recursion in the PyRv+FT method. The lowest-level nodes correspond to fastText-embedded words. As we move upward, the nodes represent combined word embeddings, capturing phrase-level meanings.



Slika 2: Vizualni prikaz piramidalne rekurzije u metodi PyRv+FT. Najniži čvorovi predstavljaju riječi u fastText vektorskoj reprezentaciji. Viši čvorovi prikazuju kombinirane vektore riječi koji obuhvaćaju značenja na razini fraza.

In this work, we leverage the Pyramidal Recursive learning (PyRv) method, introduced in our previous paper (Babić and Meštrović, 2024), to compose multiple word embeddings into unified representations. PyRv facilitates structured composition through its hierarchical learning approach, recursively combining representations at each level. Initially developed for constructing representations from tokens (subwords) up to phrases (as illustrated in Figure 1), PyRv is well-suited for combining word embeddings by progressively merging individual word vectors into phrase vectors (as is shown in Figure 2).

One of the main properties of the PyRv method is representation compositionality, which enables the composition of multiple embeddings into a coherent, semantically rich representation. This property aligns with the objective of this paper, where the focus is on effectively combining word embeddings to capture more complex linguistic structures. By recursively merging embeddings, PyRv maintains the semantic integrity of each word while constructing increasingly abstract representations at higher levels of the hierarchy.

The primary contribution of this paper is the introduction of a method for composing multi-word units into unified representations using Pyramidal Recursive learning (PyRv). To assess the effectiveness of this approach, we train the PyRv embedding model on Croatian texts and compare its performance to the widely-used baseline method of averaging word embeddings. In addition, we explore the structure of the representation space generated by PyRv's composition method. Our evaluation results validate PyRv's compositionality property.

Following this introduction, the subsequent sections of this paper are structured as follows: In Section 2,

U ovom radu koristimo metodu Pyramidal Recursive learning (PyRv), predstavljenu u našem prethodnom radu (Babić i Meštrović, 2024), kako bismo kompozicijom više vektora riječi dobili objedinjene reprezentacije. PyRv omogućuje strukturirano kombiniranje kroz hijerarhijski pristup učenju, rekurzivno kombinirajući reprezentacije na svakoj razini. Izvorno razvijena za izgradnju reprezentacija od tokena (podriječi) do fraza (kao što je prikazano na slici 1), metoda PyRv posebno je pogodna za kombiniranje vektora riječi postupnim spajanjem pojedinačnih vektora u vektorske reprezentacije fraza (kao što je prikazano na slici 2).

Jedno od glavnih svojstava metode PyRv jest kompozicionalnost reprezentacija, koja omogućuje stvaranje koherentnih, semantički bogatih reprezentacija iz više vektora. Ovo svojstvo izravno se poklapa s ciljem ovog rada, koji se fokusira na učinkovito kombiniranje vektorskih reprezentacija riječi kako bi se obuhvatile složenije jezične strukture. Rekurzivnim spajanjem vektora, PyRv održava semantički integritet svake riječi pri izgradnji reprezentacija na višim razinama hijerarhije.

Glavni doprinos ovog rada je predstavljanje metode za kompoziciju višerječnih izraza u objedinjene reprezentacije pomoću piramidalnog rekurzivnog učenja (PyRv). Kako bismo evaluirali učinkovitost ovog pristupa, treniramo PyRv model reprezentacija na hrvatskim tekstovima i uspoređujemo njegovu učinkovitost s često korištenom referentnom metodom prosječavanja vektora riječi. Također istražujemo strukturu vektorskog prostora koji proizlazi iz PyRv metode kompozicije. Rezultati evaluacije potvrđuju svojstvo kompozicionalnosti metode PyRv.

Nakon ovog uvoda, sljedeća poglavlja rada organizirani su kako slijedi: U poglavlju 2, "Povezani ra-

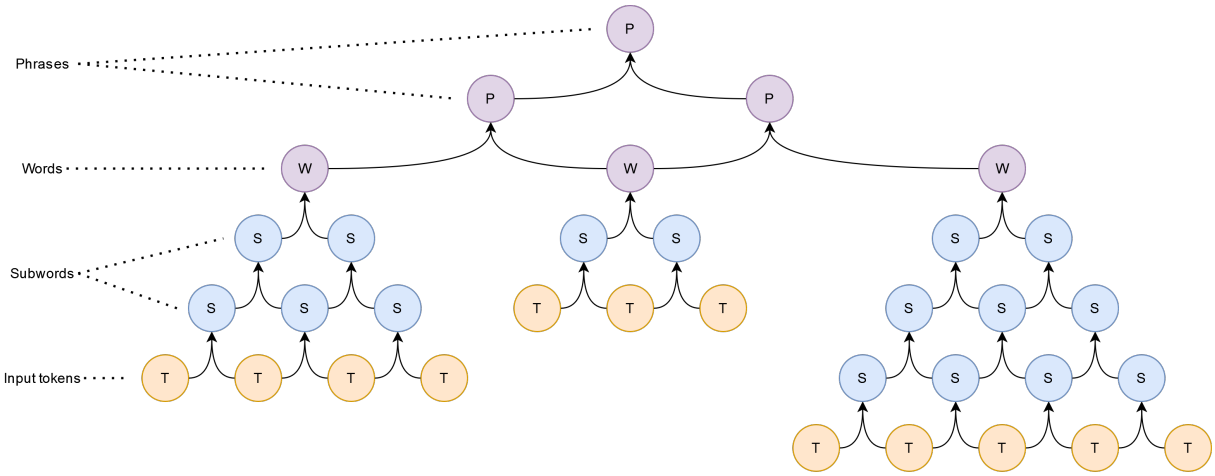


Figure 1. A visualized example of a pyramidal recursion in the PyRv method. The lowest-level nodes correspond to input tokens. Moving upward, the nodes within the three pyramids represent combined subword embeddings. At the pyramid peaks, nodes represent word embeddings, and higher nodes signify combined word embeddings, representing phrases.

Slika 1. Vizualizirani primjer piramidalne rekurzije u metodi PyRv.Čvorovi na najnižoj razini odgovaraju ulaznim tokenima. Na višim razinama unutar triju piramida nalaze se čvorovi koji predstavljaju kombinirane podriječne vektore. Na vrhovima piramida nalaze se vektori riječi, a još viši čvorovi predstavljaju kombinirane vektore riječi, tj. fraze.

"Related Work," we explore prior research, highlighting methods that compose word embeddings. Section 3, "Methodology," presents our method for composition of word embeddings. Section 4, "Evaluation," details the datasets, experiment setup, and results. Finally, in Section 5, "Conclusion," we conclude with a summary of our contributions to the field and discuss future directions.

2 Related work

Composing word embeddings to generate meaningful representations of larger text units remains a critical area of study in natural language processing. Approaches to this problem span from simple aggregation techniques that compose word embeddings to more complex neural architectures that embed entire sentences, each offering unique advantages in different contexts.

A common approach is to average word embeddings to generate a single vector representing a phrase or sentence. Averaged word embeddings were used with fastText for efficient text classification (Joulin et al., 2016). This technique, inspired by the continuous bag of words (CBOW) model (Mikolov et al., 2013), offers a computationally lightweight solution that performs competitively with deeper models in various NLP tasks.

Building upon this, an enhanced version was proposed where word embeddings are combined using weighted averages, followed by post-processing through PCA/SVD (Arora et al., 2017). The weighting scheme they propose significantly improves performance on tex-

dovi," prikazujemo dosadašnja istraživanja i metode kompozicije vektora riječi. Poglavlje 3, "Metodologija," predstavlja naš pristup kompoziciji vektorskih reprezentacija. Poglavlje 4, "Evaluacija," donosi podatkovne skupove, eksperimentalnu postavu i rezultate. Na kraju, u poglavlju 5, "Zaključak," sažimamo doprinos rada i iznosimo smjernice za budući rad.

2 Povezani radovi

Kompozicija vektorskih reprezentacija riječi s ciljem generiranja smislenih prikaza većih jezičnih jedinica ostaje ključno područje istraživanja u obradi prirodnog jezika. Pristupi ovom problemu kreću se od jednostavnih tehnika agregacije, koje kombiniraju vektore riječi, do složenijih neuronskih arhitektura koje vektoriziraju cijele rečenice, pri čemu svaki pristup ima specifične prednosti u različitim kontekstima.

Uobičajen pristup je prosječno kombiniranje vektora riječi radi generiranja jednog vektora koji predstavlja frazu ili rečenicu. Prosječni vektori riječi korišteni su s modelom fastText za učinkovitu klasifikaciju teksta (Joulin i dr., 2016). Ova tehnika, inspirirana modelom continuous bag of words (CBOW) (Mikolov i dr., 2013), nudi računalno učinkovit pristup koji u brojnim NLP zadacima konkurira dubljim modelima.

Na tome se nadograđuje poboljšana verzija u kojoj se vektori riječi kombiniraju pomoću ponderiranih prosjeka, a zatim se dodatno obrađuju metodama poput PCA/SVD (Arora i dr., 2017). Predloženi sustav ponderiranja značajno poboljšava rezultate u zadacima se-

tual similarity tasks. This method demonstrates that simple compositional techniques can rival more complex architectures, especially in unsupervised settings.

A comparative study highlighted the trade-offs between simple word averaging and more complex models like LSTMs for sentence embedding (Wieting et al., 2015). Their findings showed that while LSTMs perform well on in-domain data, simple word averaging techniques tend to outperform LSTMs in out-of-domain tasks. This suggests that straightforward compositional methods, despite their simplicity, are robust and generalizable across diverse datasets.

A recursive neural network based on syntactic parse trees was proposed to generate phrase and sentence representations, useful in tasks like sentiment analysis (Socher et al., 2013). A Self-Adaptive Hierarchical Sentence Model was introduced, using recursive structures without relying on syntax and showing that supervised learning can effectively capture compositional semantics in a non-syntactic hierarchy (Zhao et al., 2015).

Transformer (Vaswani, 2017) architectures use attention mechanism to compose embeddings. The attention mechanism was introduced in neural machine translation, allowing models to dynamically focus on different parts of the input sequence during decoding (Bahdanau, 2014). The introduction of attention helped relieve the encoder from compressing all information into a single fixed-length vector, thus enabling more flexible and effective composition of representations over sequential data.

In this work, we build on these approaches by applying the Pyramidal Recursive learning (PyRv) method (Babić and Meštrović, 2024) to recursively combine word embeddings into more abstract representations. Unlike the averaging techniques (Arora et al., 2017; Joulin et al., 2016), PyRv enables hierarchical composition, capturing both word-level and higher-level semantic structures in a more structured way. Unlike most other recursive models, which often rely on syntactic trees (Socher et al., 2013), PyRv operates without requiring explicit parse structures. Additionally, unlike hierarchical models that require supervision (Zhao et al., 2015), PyRv is fully unsupervised. Additionally, by recursively merging embeddings, our approach offers a simpler alternative to attention mechanisms for constructing rich representations of text.

3 Methodology

In this section, we introduce word embeddings and common composition techniques, followed by a detailed explanation of how the PyRv method is applied to compose word embeddings.

mantičke sličnosti teksta. Ova metoda pokazuje da jednostavne kompozicijske tehnike mogu biti usporedive sa složenijim arhitekturama, osobito u ne-nadgledanim okruženjima.

Jedno komparativno istraživanje istaknulo je kompromise između prosječnog kombiniranja riječi i složenijih modela poput LSTM-ova za rečenične reprezentacije (Wieting i dr., 2015). Rezultati pokazuju da LSTM-ovi dobro funkcioniraju na podacima iz iste domene, dok jednostavne tehnike prosjeka često nadmašuju LSTM-ove u zadacima izvan domena. To sugerira da su jednostavne kompozicijske metode, unatoč svojoj jednostavnosti, robusne i prenosive na različite skupove podataka.

Predložena je rekurzivna neuronska mreža temeljena na sintaktičkim stablima za generiranje reprezentacija fraza i rečenica, korisna u zadacima poput analize sentimenta (Socher i dr., 2013). Predstavljen je i Self-Adaptive Hierarchical Sentence Model koji koristi rekurzivne strukture bez oslanjanja na sintaksu te pokazuje da nadgledano učenje može učinkovito uhvatiti semantiku kompozicije u nesintaktičkoj hijerarhiji (Zhao i dr., 2015).

Arhitekture transformera (Vaswani, 2017) koriste mehanizam pažnje (attention) za kompoziciju reprezentacija. Taj mehanizam prvi je put uveden u kontekstu neuronskog strojnog prevođenja, gdje je modelima omogućeno da se tijekom dekodiranja dinamički fokusiraju na različite dijelove ulazne sekvence (Bahdanau, 2014). Uvođenje mehanike pažnje oslobodilo je enkoder potrebe da sve informacije sažme u jedan vektor fiksne duljine, čime se omogućila fleksibilnija i učinkovitija kompozicija reprezentacija nad sekvencijalnim podacima.

U ovom radu gradimo na spomenute pristupe koristeći metodu piramidalnog rekurzivnog učenja (PyRv) (Babić i Meštrović, 2024) kako bismo rekurzivno kombinirali vektorske reprezentacije riječi. Za razliku od metoda prosjeka (Arora i dr., 2017; Joulin i dr., 2016), PyRv omogućuje hijerarhijsku kompoziciju, uzimajući u obzir semantičke strukture na razini riječi i višim razinama na strukturiraniji način. Za razliku od većine drugih rekurzivnih modela, koji se često oslanjaju na sintaktička stabla (Socher i dr., 2013), PyRv ne zahtijeva eksplicitne strukture za parsiranje. Također, za razliku od hijerarhijskih modela koji zahtijevaju nadgledano učenje (Zhao i dr., 2015), PyRv je potpuno nenadgledan. Nadalje, rekurzivnim spajanjem reprezentacija, naš pristup nudi jednostavniju alternativu mehanizmima pažnje u konstrukciji semantički bogatih reprezentacija teksta.

3 Metodologija

U ovom poglavlju predstavljamo vektorske reprezentacije riječi (eng. "word embeddings") i uobičajene tehnike kompozicije, nakon čega slijedi detaljno objašnjenje kako se metoda piramidalnog rekurzivnog učenja

3.1 Word embedding

Word embeddings, such as those produced by fastText (Bojanowski et al., 2017), are commonly used to represent individual words. To represent multiple words, basic techniques like averaging or concatenation can be applied. However, both approaches come with notable limitations.

When averaging embeddings (e.g., taking the mean of multiple word vectors), important information, particularly word order, is lost. Concatenation preserves all information but presents two significant challenges.

First, concatenated embeddings vary in dimensionality depending on the number of words, which complicates their use as input for models that require a fixed input size. Second, the dimensionality of concatenated representations can grow excessively large. For instance, a 5-word phrase embedded with 300-dimensional fastText results in a 1500-dimensional vector (5x300).

In our evaluation, we use averaging to compose Croatian fastText embeddings (Grave et al., 2018) to maintain a consistent dimensionality across all methods, ensuring that the results are comparable.

3.2 PyRv with fastText word embeddings

Pyramidal Recursive learning (PyRv) is a method designed to construct hierarchical representations of text, moving progressively from low-level units such as characters or subwords to higher-level representations such as words, phrases, sentences, and even paragraphs. PyRv combines representations recursively, forming increasingly abstract and semantically rich embeddings at each level of the hierarchy.

To address the limitations of averaging and concatenation, we explore the use of PyRv for composing multiple word embeddings into a single, unified representation. Unlike our previous work on PyRv (Babić and Meštrović, 2024), where the recursion starts from subwords or tokens, in this study we begin with word embeddings produced by fastText. This hybrid approach is referred to as PyRv+FT.

For this paper, we use pre-trained Croatian fastText word vectors (Grave et al., 2018) to embed words, which are then recursively combined into phrase embeddings via the PyRvNN model. The PyRv+FT embeddings are trained on Croatian Wikipedia texts (Wikimedia public dump, January 11, 2020) for 10 epochs.

We evaluate PyRv+FT on two downstream tasks, described in detail in the next section. Through this process, we investigate how PyRv improves the com-

(eng. "Pyramidal Recursive learning", PyRv) primjenjuje za kompoziciju vektorskih reprezentacija riječi.

3.1 Vektorske reprezentacije riječi

Vektorske reprezentacije riječi (eng. "word embeddings"), poput onih koje generira fastText (Bojanowski i dr., 2017), često se koriste za reprezentaciju pojedinačnih riječi. Za reprezentaciju više riječi mogu se primijeniti osnovne tehnike poput prosjeka ili konkatenacije. Međutim, obje metode imaju značajna ograničenja.

Kod prosjeka (npr. računanja srednje vrijednosti više vektora riječi), gubi se važna informacija, posebice redoslijed riječi. Konkatenacija zadržava sve informacije, ali donosi dva velika izazova.

Prvo, vektori dobiveni konkatenacijom variraju u dimenzionalnosti ovisno o broju riječi, što otežava njihovu primjenu kao ulaz za modele koji zahtijevaju fiksnu veličinu ulaza. Drugo, dimenzionalnost konkateniranih reprezentacija može postati pretjerano velika. Na primjer, fraza od 5 riječi vektorizirana pomoću fastText-a s 300 dimenzija rezultira vektorom od 1500 dimenzija (5x300).

U našoj evaluaciji koristimo metodu prosjeka za kompoziciju hrvatskih fastText vektora riječi (Grave i dr., 2018) kako bismo održali konzistentnu dimenzionalnost među svim metodama i time omogućili usporedivost rezultata.

3.2 PyRv s fastText vektorskim reprezentacijama riječi

Piramidalno rekurzivno učenje (eng. "Pyramidal Recursive learning", PyRv) je metoda osmišljena za izgradnju hijerarhijskih reprezentacija teksta, koja progresivno prelazi s nižih razina poput znakova ili podriječi na više razine poput riječi, fraza, rečenica pa čak i odlomaka. PyRv rekurzivno kombinira reprezentacije, formirajući sve apstraktnije i semantički bogate vektore na svakoj razini hijerarhije.

Kako bismo zaobišli ograničenja prosjeka i konkatenacije, istražujemo primjenu metode PyRv za kompoziciju više vektorskih reprezentacija riječi u jednu objedinjenu reprezentaciju. Za razliku od našeg prethodnog rada o metodi PyRv (Babić i Meštrović, 2024), gdje rekurzija započinje s podriječima ili tokenima, u ovom istraživanju polazimo od vektorskih reprezentacija riječi generiranih pomoću fastText-a. Ovaj hibridni pristup nazivamo PyRv+FT.

U ovom radu koristimo unaprijed istrenirane hrvatske fastText vektore riječi (Grave i dr., 2018) za vektorizaciju riječi, koje se zatim rekurzivno kombiniraju u frazne reprezentacije pomoću modela PyRvNN.
PyRv+FT vektori trenirani su na tekstovima s hrvatske Wikipedije (Wikimedia javni podaci, 11. siječnja 2020.) kroz 10 epoha.

Evaluiramo PyRv+FT na dva zadatka krajnje pri-

positionality of word embeddings, while maintaining manageable dimensionality and enhancing representational quality.

3.3 Mathematical Formulation

To formalize the distinction between averaging and our proposed PyRv+FT approach, we define their respective composition functions. Let a phrase P be a sequence of n words, $P = (w_1, w_2, \ldots, w_n)$. Each word w_i is mapped to a d-dimensional vector embedding $\boldsymbol{v}_i \in \mathbb{R}^d$ using fastText. The goal is to compose the sequence of vectors $(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n)$ into a single phrase embedding $\boldsymbol{e}_P \in \mathbb{R}^d$.

Averaging. The averaging method computes the phrase embedding $\boldsymbol{e}_P^{\text{avg}}$ by taking the element-wise mean of the constituent word vectors:

$$\boldsymbol{e}_P^{\text{avg}} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{v}_i \tag{1}$$

The primary limitation of this model is apparent from its formulation. Since vector addition is commutative ($\boldsymbol{v}_i + \boldsymbol{v}_j = \boldsymbol{v}_j + \boldsymbol{v}_i$), the resulting embedding $\boldsymbol{e}_P^{\text{avg}}$ is invariant to the order of words in the phrase, losing crucial syntactic and semantic information.

Pyramidal Recursive Composition (PyRv+FT). The PyRv+FT method constructs the phrase embedding $\boldsymbol{e}_P^{\text{PyRv}}$ through a hierarchical and recursive process. The composition starts at the lowest level (level 0), which consists of the initial word embeddings $(\boldsymbol{v}_1^{(0)}, \boldsymbol{v}_2^{(0)}, \ldots, \boldsymbol{v}_n^{(0)})$.

At each subsequent level k, adjacent pairs of vectors from level $k-1$ are combined using a parameterized composition function f_θ (the PyRvNN model) to form a new, higher-level representation. This process is repeated until a single vector remains. A new vector $\boldsymbol{v}_i^{(k)}$ at level k is generated from level $k-1$ as follows:

$$\boldsymbol{v}_i^{(k)} = f_\theta(\boldsymbol{v}_i^{(k-1)}, \boldsymbol{v}_{i+1}^{(k-1)}) \tag{2}$$

The final phrase embedding $\boldsymbol{e}_P^{\text{PyRv}}$ is the single vector remaining at the top of the pyramid.

Unlike averaging, the composition function f_θ is non-commutative because concatenation is order-dependent (i.e., $[\boldsymbol{a}; \boldsymbol{b}] \neq [\boldsymbol{b}; \boldsymbol{a}]$). This allows the model to capture word order. Furthermore, the learnable parameters θ enable the model to learn complex compositional semantics from data, offering a significant advantage over the static, order-agnostic nature of averaging.

mjene, detaljno opisana u sljedećem poglavlju. Kroz ovaj proces istražujemo kako PyRv poboljšava kompozicionalnost vektorskih reprezentacija riječi, dok istovremeno održava prihvatljivu dimenzionalnost i poboljšava kvalitetu reprezentacije.

3.3 Matematička formulacija

Kako bismo formalizirali razliku između metode prosjeka i našeg predloženog PyRv+FT pristupa, definiramo njihove odgovarajuće kompozicijske funkcije. Neka je fraza P niz od n riječi, $P = (w_1, w_2, \ldots, w_n)$. Svaka riječ w_i preslikava se u d-dimenzionalnu vektorsku reprezentaciju $\boldsymbol{v}_i \in \mathbb{R}^d$ pomoću fastTexta. Cilj je sastaviti niz vektora $(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n)$ u jedinstvenu vektorsku reprezentaciju fraze $\boldsymbol{e}_P \in \mathbb{R}^d$.

Prosjek. Metoda prosjeka računa vektorsku reprezentaciju fraze $\boldsymbol{e}_P^{\text{avg}}$ uzimanjem srednje vrijednosti po elementima pripadajućih vektora riječi:

$$\boldsymbol{e}_P^{\text{avg}} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{v}_i \tag{1}$$

Glavno ograničenje ovog modela vidljivo je iz njegove formulacije. Budući da je zbrajanje vektora komutativno ($\boldsymbol{v}_i + \boldsymbol{v}_j = \boldsymbol{v}_j + \boldsymbol{v}_i$), rezultirajuća reprezentacija $\boldsymbol{e}_P^{\text{avg}}$ je neosjetljiva na redoslijed riječi u frazi, čime se gube bitne sintaktičke i semantičke informacije.

Piramidalna rekurzivna kompozicija (PyRv+FT). PyRv+FT metoda konstruira vektorsku reprezentaciju fraze $\boldsymbol{e}_P^{\text{PyRv}}$ kroz hijerarhijski i rekurzivni proces. Kompozicija započinje na najnižoj razini (razina 0), koja se sastoji od početnih vektorskih reprezentacija riječi $(\boldsymbol{v}_1^{(0)}, \boldsymbol{v}_2^{(0)}, \ldots, \boldsymbol{v}_n^{(0)})$.

Na svakoj sljedećoj razini k, susjedni parovi vektora s razine $k-1$ kombiniraju se pomoću parametrizirane kompozicijske funkcije f_θ (model PyRvNN) kako bi se oblikovala nova reprezentacija više razine. Ovaj proces se ponavlja dok ne preostane samo jedan vektor. Novi vektor $\boldsymbol{v}_i^{(k)}$ na razini k generira se iz razine $k-1$ na sljedeći način:

$$\boldsymbol{v}_i^{(k)} = f_\theta(\boldsymbol{v}_i^{(k-1)}, \boldsymbol{v}_{i+1}^{(k-1)}) \tag{2}$$

Konačna vektorska reprezentacija fraze $\boldsymbol{e}_P^{\text{PyRv}}$ je jedinstveni vektor koji preostaje na vrhu piramide.

Za razliku od metode prosjeka, kompozicijska funkcija f_θ nije komutativna jer konkatenacija ovisi o redoslijedu (tj. $[\boldsymbol{a}; \boldsymbol{b}] \neq [\boldsymbol{b}; \boldsymbol{a}]$). To omogućuje modelu da zabilježi redoslijed riječi. Nadalje, učljivi parametri θ omogućuju modelu učenje složene kompozicijske semantike iz podataka, što nudi značajnu prednost u odnosu na statičnu prirodu prosjeka koja je neosjetljiva na redoslijed.

4 Evaluation

In this section, we describe the evaluation of fastText and PyRv+FT embeddings on two important NLP tasks: Universal Part-of-Speech tagging (UPOS) and Universal Dependency Relation labeling (DEPREL). UPOS tags represent core grammatical categories such as nouns, verbs, and adjectives, while DEPREL captures the syntactic relationships between words in a sentence, indicating dependencies like subjects, objects, and modifiers.

For this evaluation, we use the hr500k 2.0 dataset (Ljubešić et al., 2016), a Croatian corpus with labeled data for both UPOS and DEPREL tasks (amongst others). This dataset contains 901 texts, 24,763 sentences, and a total of 499,635 tokens.

Additionally, we perform a qualitative analysis of PyRv+FT embeddings by visualizing the representation space to gain deeper insights into its structure and characteristics.

4.1 Experiment setup

To assess the performance of different embedding methods on the downstream tasks of UPOS and DEPREL, we use a multi-layer perceptron (MLP) model with one hidden layer. The hidden layer consists of 1,000 neurons and uses the ReLU activation function. The input to the model is a 300-dimensional vector (the size of both fastText and PyRv+FT embeddings). The output layer, with softmax activation, adjusts to the number of classes in each task: 17 classes for UPOS and 37 classes for DEPREL. Each evaluation is conducted by training the MLP for one epoch.

The embedding procedure remains consistent across both UPOS and DEPREL tasks, differing only in the MLP output labels.

The method of embedding a word from a sentence depends on the embedding strategy employed:

- **fastText 1 word**: Embeds only the target word, ignoring its surrounding context.
- **mean fastText N words**: Represents the target word by embedding all N words (the target word and its N-1 neighboring context words) separately and averaging the embeddings to obtain a final representation.
- **PyRv+FT N words**: Embeds each word using fastText, but instead of averaging the N embeddings, it recursively combines them using the PyRvNN model to generate a single, unified embedding.

4.2 Quantitative results

UPOS. Part-of-speech tagging is a relatively simple task where the surrounding word context does not provide significant benefits for classification. We include UPOS evaluation primarily to demonstrate how aver-

4 Evaluacija

U ovom poglavlju opisujemo evaluaciju fastText i PyRv+FT vektorskih reprezentacija na dva bitna zadatka obrade prirodnog jezika: označavanje univerzalnih vrsta riječi (UPOS) i označavanje univerzalnih sintaktičkih odnosa (DEPREL). Oznake UPOS predstavljaju osnovne gramatičke kategorije poput imenica, glagola i pridjeva, dok DEPREL bilježi sintaktičke odnose među riječima u rečenici, poput subjekata, objekata i atributa.

Za ovu evaluaciju koristimo skup podataka hr500k 2.0 (Ljubešić i dr., 2016), hrvatski korpus s anotiranim podacima za UPOS i DEPREL zadatke (među ostalim). Skup podataka sadrži 901 tekst, 24.763 rečenice i ukupno 499.635 tokena.

Dodatno, provodimo kvalitativnu analizu PyRv+FT reprezentacija vizualizacijom njihovog reprezentacijskog prostora, kako bismo dobili dublji uvid u njegovu strukturu i obilježja.

4.1 Postavke eksperimenta

Za procjenu performansi različitih metoda vektorskog reprezentiranja na zadacima UPOS i DEPREL koristimo višeslojni perceptron (MLP) model s jednom skrivenom slojem. Skriveni sloj sadrži 1.000 neurona i koristi ReLU aktivacijsku funkciju. Ulaz u model je 300-dimenzionalni vektor (što odgovara dimenziji fastText i PyRv+FT vektora). Izlazni sloj koristi softmax aktivaciju i prilagođava se broju klasa svakog zadatka: 17 za UPOS i 37 za DEPREL. Svaka evaluacija provodi se treniranjem modela MLP kroz jednu epohu.

Postupak vektorizacije ostaje jednak za UPOS i DEPREL, a razlikuju se samo oznake izlaznih klasa.

Način vektorizacije riječi iz rečenice ovisi o korištenoj strategiji:

- **fastText 1 riječ**: Vektorizira samo ciljnu riječ, zanemarujući njezin kontekst.
- **prosjek fastText N riječi**: Reprezentira ciljnu riječ tako da se zasebno vektoriziraju sve N riječi (ciljna i N-1 susjedne) i zatim izračuna prosjek.
- **PyRv+FT N riječi**: Svaka riječ se vektorizira fastTextom, ali se umjesto prosjeka tih N vektora koristi model PyRvNN koji ih rekurzivno kombinira u jednu objedinjenu reprezentaciju.

4.2 Kvantitativni rezultati

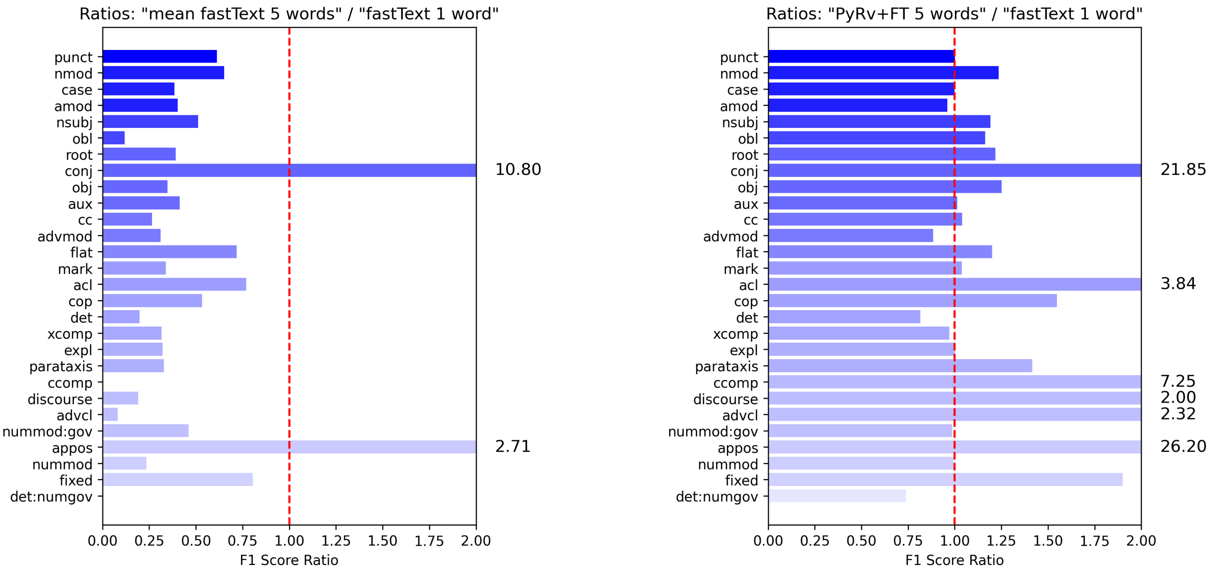
UPOS. Označavanje vrsta riječi relativno je jednostavan zadatak u kojem kontekst susjednih riječi ne donosi velike prednosti. UPOS evaluaciju uključujemo kako bismo pokazali da prosječna fastText reprezenta-

Table 1. UPOS results, Macro and Weighted averages. **Tablica 1**. Rezultati UPOS-a, makro i ponderirani prosjeci.

	Accuracy	**Precision**		**Recall**		**F1 score**	
		M. avg	W. avg	M. avg	W. avg	M. avg	W. avg
fastText 1 word	**0.95**	**0.91**	**0.95**	**0.89**	**0.95**	0.89	**0.95**
mean fastText 3 words	0.61	0.57	0.61	0.59	0.61	0.57	0.61
PyRv+FT 3 words	0.93	0.9	0.93	**0.89**	0.93	**0.9**	0.93

Table 2. DEPREL results, Macro and Weighted averages. **Tablica 2**. Rezultati DEPREL-a, makro i ponderirani prosjeci.

	Accuracy	**Precision**		**Recall**		**F1 score**	
		M. avg	W. avg	M. avg	W. avg	M. avg	W. avg
fastText 1 word	0.71	0.52	0.68	0.48	0.71	0.47	0.68
mean fastText 5 words	0.34	0.25	0.34	0.19	0.34	0.19	0.31
PyRv+FT 5 words	**0.77**	**0.58**	**0.77**	**0.55**	**0.77**	**0.56**	**0.76**



(a) "mean fastText 5 words" F1 scores divided by "fastText 1 word" F1 scores.
(a) Omjer F1 rezultata za "prosjek fastText 5 riječi" u odnosu na "fastText 1 riječ".

(b) "PyRv+FT 5 words" F1 scores divided by "fastText 1 word" F1 scores.
(b) Omjer F1 rezultata za "PyRv+FT 5 riječi" u odnosu na "fastText 1 riječ".

Figure 3. DEPREL relative F1 score ratios (by class) comparing different composition methods. The left plot (a) shows the ratio of F1 scores for "mean fastText 5 words" versus "fastText 1 word", while the right plot (b) compares "PyRv+FT 5 words" versus "fastText 1 word". Each bar represents a class, and the length of the bar indicates the relative performance of the model. Classes are ordered by support value in the test set (larger on the top).
Slika 3. Relativni F1 omjeri (po klasi) za DEPREL, uspoređujući metode kompozicije. Lijevi graf (a) prikazuje omjer F1 rezultata za "prosjek fastText 5 riječi" prema "fastText 1 riječ", dok desni graf (b) prikazuje odnos za "PyRv+FT 5 riječi". Svaka traka predstavlja klasu, a njezina duljina pokazuje relativne performanse. Klase su poredane po zastupljenosti u testnom skupu (najzastupljenije na vrhu).

aging fastText word embeddings can degrade downstream performance, while combining fastText word embeddings using PyRvNN preserves much of the embedding quality.

When using fastText to embed a single word with-

cija može narušiti rezultate, dok kombiniranje fastText vektora putem PyRvNN-a zadržava većinu kvalitete reprezentacije.

Kada se koristi fastText za vektorizaciju samo jedne riječi bez konteksta, postiže se točnost od 95%. Me-

out considering its context, we achieve an accuracy of 95%. However, averaging fastText embeddings over three words leads to a substantial drop in performance, with accuracy falling to 61%. In contrast, when combining fastText embeddings for three words using PyRvNN, the performance degradation is minimized, yielding an accuracy of 93%. These results highlight how PyRvNN can effectively mitigate the loss of information that occurs when averaging word embeddings. Detailed results are shown in Table 1.

DEPREL. The dependency relation task is more complex than UPOS, as it requires understanding the syntactic relationships between words. In this case, enriching word embeddings with surrounding context can significantly improve classification performance.

When embedding a single word using fastText (without its context), the model achieves an accuracy of 71%. However, averaging five fastText word embeddings, including the target word and its four neighbors, results in a sharp decline in performance, with accuracy dropping to 34%. This reduction in accuracy reflects how averaging word embeddings leads to the loss of important information, including word order and syntactic structure. By contrast, composing five fastText word embeddings using PyRvNN boosts accuracy to 77%, demonstrating the method's ability to capture more nuanced relationships between words.

Evaluation results are presented in Table 2 with more detailed results (by class) available in Tables 5, 6, and 7 in the Appendix.

Figure 3 presents bar plots comparing F1 scores between two composition methods, mean averaging and PyRv, relative to single word fastText embeddings' performance. The left plot 3a illustrates the F1 score ratio between "mean fastText 5 words" and "fastText 1 word", while the right plot 3b contrasts "PyRv+FT 5 words" with "fastText 1 word".

In the following analysis, we focus on three notable classes: punctuation, conjuncts, and adnominal clauses. These were selected because punctuation classification does not rely on context, classifying conjuncts without context is nearly impossible, and for adnominal clauses, context is helpful but averaging tends to degrade performance.

Punctuation (punct) refers to punctuation marks such as ".", "?", "!", and ",". Since punctuation is straightforward to classify without context, a single fastText embedding for the target word alone achieves a perfect F1 score of 1. Averaging five fastText embeddings (the target word and its four neighboring words) significantly degrades performance, reducing the F1 score to 0.61. However, using PyRv to compose these embeddings preserves the high performance, maintaining an F1 score of 1.

Conjunct (conj) denotes a relation between elements connected by coordinating conjunctions like "and," "or," or ",". In coordinate structures, the first element is conventionally treated as the head, with subsequent el-

đutim, prosjek triju fastText vektora dovodi do značajnog pada performansi – točnost pada na 61%. Suprotno tome, kada se isti vektori kombiniraju koristeći PyRvNN, pad performansi je znatno manji – točnost iznosi 93%. Ovi rezultati pokazuju kako PyRvNN učinkovito ublažava gubitak informacija do kojeg dolazi kada se računa prosjek. Detaljni rezultati prikazani su u Tablici 1.

DEPREL. Zadatak označavanja sintaktičkih odnosa među riječima u rečenici znatno je složeniji jer zahtijeva razumijevanje odnosa među riječima. U ovom slučaju, obogaćivanje vektora kontekstom može značajno unaprijediti klasifikaciju.

Korištenjem fastText-a za jednu riječ bez konteksta postiže se točnost od 71%. No, kada se koristi prosjek pet fastText vektora (ciljna riječ i četiri susjedne), točnost drastično pada na 34%. Taj pad ukazuje na to da se prosjekom gubi važne informacije – poput redoslijeda riječi i sintaktičke strukture. S druge strane, PyRv kompozicija pet vektora rezultira točnošću od 77%, što potvrđuje da metoda bolje zahvaća dublje odnose među riječima.

Rezultati evaluacije prikazani su u Tablici 2, dok su detaljni rezultati po klasama dostupni u Tablicama 5, 6 i 7 u Dodatku.

Na Slici 3 prikazani su grafikoni s usporedbama F1 rezultata između metoda prosjeka i PyRv kompozicije u odnosu na fastText za jednu riječ. Lijevi prikaz 3a prikazuje omjer za "prosjek fastText 5 riječi", a desni 3b za "PyRv+FT 5 riječi".

U sljedećoj analizi fokusiramo se na tri istaknute klase: interpunkciju, koordinirane veznike i odnosne surečenice. Odabrane su jer klasifikacija interpunkcije ne ovisi o kontekstu, klasificiranje koordiniranih veznika bez konteksta gotovo je nemoguće, dok je za odnosne surečenice kontekst koristan, ali prosječno kombiniranje vektora narušava performanse.

Interpunkcija (punct) odnosi se na interpunkcijske znakove poput ".", "?", "!" i ",". Budući da je interpunkciju jednostavno klasificirati bez konteksta, korištenje fastText reprezentacije jedne riječi postiže savršen F1 rezultat od 1. Međutim, prosječno kombiniranje pet fastText reprezentacija (ciljna riječ i četiri susjedne) znatno narušava performanse, smanjujući F1 rezultat na 0,61. S druge strane, kompozicija ovih reprezentacija pomoću metode PyRv zadržava visoku točnost, održavajući F1 rezultat na 1.

Konjunktor (conj) označava relaciju između elemenata povezanih koordinacijskim veznicima poput "i", "ili" ili ",". U koordinacijskim strukturama, prvi element konvencionalno se tretira kao glava, a ostali se

ements connected through the conj relation. For example, in the sentence "Bill is *big* and *honest*," the word "honest" is labeled as conj (connected to "big"). Similarly, in "He *came* home, *took* a shower and immediately *went* to bed," the words "took" and "went" are both labeled as conj (connected to "came"). Classifying conjuncts accurately requires contextual information. A single fastText word embedding, without any context, yields a poor F1 score of 0.03. Averaging the embeddings of the target word and its four neighbors improves performance significantly, achieving an F1 score of 0.32 (a 10.8x improvement). Composing these embeddings using PyRv further boosts performance, reaching an F1 score of 0.64 (a 21.85x improvement over the single word embedding).

Adnominal clause (acl) refers to finite or non-finite clauses that modify a nominal. For instance, in "the *issues* as he *sees* them," the word "sees" is labeled as acl (connected to "issues"). In "There are many online *sites offering* booking facilities," the word "offering" is labeled as acl (connected to "sites"). Using a single fastText word embedding results in an F1 score of 0.14. Averaging the target word's embedding with its four neighbors actually degrades performance slightly, reducing the F1 score to 0.11. In contrast, composing these word embeddings with PyRv substantially improves performance, raising the F1 score to 0.53 (a 3.84x improvement over the single word embedding).

4.3 Qualitative analysis

To gain qualitative insights into the structure of PyRv+FT embeddings, we visualize the representation space. A portion of this space is shown in Figures 4 and 5 (in the Appendix). In these visualizations, each node represents a phrase consisting of two or three words. A two-word phrase is connected to a three-word phrase if the shorter phrase is part of the longer one.

We highlight specific clusters within the visualized space, with detailed examples of phrases from these clusters (translated to English) provided in Tables 3 and 4. The areas circled in the figures contain phrases built around the prepositions "u" (Croatian for "in") and "na" (Croatian for "on"). For example, Area C contains two-word phrases like "primjena **na**" (eng. "application on"), while Area A includes phrases such as "**na** svijet" (eng. "on the world"). Similarly, Area B features three-word phrases like "primjena **na** svijet" (eng. "application on the world"), and Area D includes phrases like "**na** svijet oko" (eng. "on the world around"). Same holds for the preposition "u".

The organization of phrases in the representation space is not random: phrases with similar syntactic structures (e.g., where the preposition appears at the beginning, middle, or end of the phrase) tend to cluster together. Furthermore, within these broader areas, smaller sub-clusters form based on the specific prepo-

povezuju conj relacijom. Na primjer, u rečenici "Bill je *velik* i *pošten*", riječ "pošten" označena je kao conj (povezana s "velik"). Slično, u "On je *došao* kući, *istuširao* se i odmah *otišao* na spavanje," riječi "istuširao" i "otišao" označene su kao conj (povezane s "došao"). Točna klasifikacija conj relacije zahtijeva kontekstualne informacije. FastText reprezentacija jedne riječi bez konteksta daje vrlo loš F1 rezultat od 0,03. Prosječno kombiniranje reprezentacija ciljne riječi i četiri susjeda znatno poboljšava performanse, postižući F1 rezultat od 0,32 (10,8 puta bolje). Kompozicija istih reprezentacija pomoću metode PyRv dodatno povećava točnost na 0,64 (21,85 puta bolje u odnosu na pojedinačnu riječ).

Atributna rečenica (acl) odnosi se na finitne ili nefinitne zavisne rečenice koje pobliže određuju ili opisuju imensku riječ. Na primjer, u rečenici "*problemi* kako ih on *vidi*", riječ "vidi" označena je kao acl (povezana s "problemi"). U rečenici "Postoji mnogo internetskih *stranica* koje *nude* mogućnost rezervacije," riječ "nude" označena je kao acl (povezana sa "stranica"). Korištenje fastText reprezentacije jedne riječi rezultira F1 rezultatom od 0,14. Prosječno kombiniranje s četiri susjeda zapravo dodatno narušava performanse, smanjujući F1 rezultat na 0,11. Nasuprot tome, kompozicija tih reprezentacija pomoću metode PyRv značajno poboljšava performanse, povećavajući F1 rezultat na 0,53 (3,84 puta bolje u odnosu na pojedinačnu riječ).

4.3 Kvalitativna analiza

Kako bismo dobili kvalitativni uvid u strukturu PyRv+FT reprezentacija, vizualiziramo njihov reprezentacijski prostor. Dio tog prostora prikazan je na Slikama 4 i 5 (u Dodatku). U tim vizualizacijama svaki čvor predstavlja frazu od dvije ili tri riječi. Fraza od dvije riječi povezana je s frazom od tri riječi ako je kraća fraza podskup duže.

Ističemo određene klastere unutar prikazanog prostora, s detaljnim primjerima fraza iz tih klastera (prevedenima na engleski) prikazanima u Tablicama 3 i 4. Područja zaokružena na slikama sadrže fraze izgrađene oko prijedloga "u" (hrvatski za "in") i "na" (hrvatski za "on"). Na primjer, Područje C sadrži dvočlane fraze poput "primjena **na**" (engl. "application on"), dok Područje A uključuje izraze poput "**na** svijet" (engl. "on the world"). Slično tome, Područje B sadrži tročlane fraze poput "primjena **na** svijet" (engl. "application on the world"), a Područje D uključuje fraze poput "**na** svijet oko" (engl. "on the world around"). Isti obrazac vrijedi i za fraze s prijedlogom "u".

Organizacija fraza u reprezentacijskom prostoru nije nasumična: fraze sa sličnom sintaktičkom strukturom (npr. prijedlog na početku, u sredini ili na kraju fraze) sklone su grupiranju. Nadalje, unutar šireg područja pojavljuju se manji podklasteri koji se dodatno organiziraju prema tome sadrži li fraza prijedlog "u" ili "na".

sition ("u" or "na") present in the phrase.

Table 3. Phrases by areas (A, B, and C) in the PyRv+FT representation space, translated to English (some phrases are longer when translated).

Preposition "on" (cro. "na")		
Area C	**Area B**	**Area A**
application on	application on the world	on the world
are on	are on local	on local
assistant on	assistant on the subject	on the subject
media on	media on protest	on the protest
vat on	vat on tickets	on tickets
based on	based on data	on data
finance on	finance on revenues	on revenues
os on	os on which	on which
dollars on	dollars on google	on google
found on	found on the third	on the third
relation on	relation on the past	on the past
Preposition "in" (cro. "u")		
Area C	**Area B**	**Area A**
enthusiast in	enthusiast in the river	in the river
currently in	currently in testing	in testing
released in	released in circulation	in circulation
circulation in	circulation in June	in June
by activity in	by activity in teaching	in teaching
work in	work in the wood	in the wood
tickets in	tickets in europe	in europe
musician in	musician in croatia	in croatia
only in	only in the past	in the past
drop in	drop in the sea	in the sea
is in	is in the past	in the past
year in	year in croatia	in croatia
percent in	percent in relation	in relation
and in	and in the average	in the average
. in	. in this	in this

Tablica 3. Izrazi po područjima (A, B i C) u PyRv+FT prostoru reprezentacije (izvorni hrvatski izrazi).

Prijedlog "na" (engl. "on")		
Područje C	**Područje B**	**Područje A**
primjena na	primjena na svijet	na svijet
su na	su na lokalnim	na lokalnim
asistentent na	asistentent na predmetu	na predmetu
medije na	medije na prosvjed	na prosvjed
pdv-a na	pdv-a na ulaznice	na ulaznice
baziranim na	baziranim na podacima	na podacima
financija na	financija na prihodima	na prihodima
os na	os na koji	na koji
dolara na	dolara na google	na google
nalazi na	nalazi na trećoj	na trećoj
odnosu na	odnosu na prošlu	na prošlu
Prijedlog "u" (engl. "in")		
Područje C	**Područje B**	**Područje A**
entuzijasta u	entuzijasta u rijeci	u rijeci
trenutno u	trenutno u testiranju	u testiranju
puštena u	puštena u opticaj	u opticaj
opticaj u	opticaj u lipnju	u lipnju
aktivnošću u	aktivnošću u nastavi	u nastavi
rada u	rada u drvnoj	u drvnoj
ulaznice u	ulaznice u europi	u europi
glazbenika u	glazbenika u hrvatskoj	u hrvatskoj
samo u	samo u protekle	u protekle
kap u	kap u moru	u moru
se u	se u protekloj	u protekloj
godini u	godini u hrvatskoj	u hrvatskoj
posto u	posto u odnosu	u odnosu
i u	i u prosjeku	u prosjeku
. u	. u ovoj	u ovoj

Table 4. Phrases in area D in the PyRv+FT representation space, translated to English (some phrases are longer when translated).

Area D	
Preposition "on" (cro. "na")	**Preposition "in" (cro. "u")**
on the world around	in the river which
on local elections	in testing and
on the subject of organization	in circulation in
on the protest of musicians	in june 2009
on tickets for	in teaching 1982
on the data collected	in the wood industry
on budget revenues	in europe .
on which this	in croatia only
on google play	in the past two
on the third position	in the sea of state
on the past year	in the past year
	in croatia is not
	in relation to
	in the average spends
	in this praise

Tablica 4. Izrazi u području D u PyRv+FT prostoru reprezentacije (izvorni hrvatski izrazi).

Područje D	
Prijedlog "na" (engl. "on")	**Prijedlog "u" (engl. "in")**
na svijet oko	u rijeci koji
na lokalnim izborima	u testiranju i
na predmetu organizacije	u opticaj u
na prosvjed glazbanika	u lipnju 2009.
na ulaznice u	u nastavi 1982.
na podacima prikupljenim	u drvnoj industriji
na prihodima proračuna	u europi .
na koji ova	u hrvatskoj samo
na google play	u protekle dvije
na trećoj poziciji	u moru državnog
na prošlu godinu	u protekloj godini
	u hrvatskoj nije
	u odnosu na
	u prosjeku troši
	u ovoj hvale

5 Conclusion

In this paper, we explored the use of Pyramidal Recursive learning (PyRv) for the composition of word embeddings and evaluated its ability to generate meaningful representations of multi-word units. Our findings show that PyRv outperforms simple averaging methods in embedding composition.

In the part-of-speech tagging task, where word context is less crucial, single word embeddings achieve an accuracy of 95%. Averaging 3-word context embeddings reduces this to 61% due to the loss of word order information, while PyRv retains a high accuracy of 93% by effectively preserving word order. In the more complex task of dependency relation labeling, where single word embeddings reach 71% accuracy, averaging embeddings for 5-word contexts results in a sharp decline to 34%. In contrast, composing context words with PyRv attains a significantly higher accuracy of 77%, demonstrating its superior capability in integrating multiple word embeddings into cohesive and expressive representations. This validates the effectiveness of the compositionality property of PyRv in real-world tasks.

The primary contribution of this work is the introduction of a method for composing multi-word units into unified representations using PyRv. We provided an evaluation of its effectiveness compared to averaging word embeddings and validated its compositionality property. Additionally, we explored the structure of the representation space generated by PyRv's compositional approach. By training the PyRv model on Croatian texts, we demonstrated its flexibility and potential for application across diverse languages.

Future work could further probe PyRv's effectiveness as a compositional model. A natural next step would be its evaluation on standard English benchmarks for tasks highly sensitive to semantic composition. This would also enable a direct comparison against other composition strategies, particularly those employed by state-of-the-art transformer-based models. Investigating architectural variations, such as enhancing the recursive function with attention mechanisms, also presents a promising research avenue.

Acknowledgments

This work has been fully supported by the University of Rijeka project uniri-iskusni-drustv-23-95.

5 Zaključak

U ovom radu istražili smo primjenu piramidalnog rekurzivnog učenja (PyRv) za kompoziciju vektorskih prikaza riječi te smo evaluirali njegovu sposobnost generiranja smislenih reprezentacija višerječnih izraza. Naši rezultati pokazuju da PyRv nadmašuje jednostavne metode prosjeka u zadacima kompozicije vektora.

U zadatku označavanja vrsta riječi, gdje kontekst ima manju ulogu, pojedinačni vektori riječi postižu točnost od 95%. Vektor dobiven metodom prosjeka konteksta sa 3 riječi značajno smanjuje točnost na 61% zbog gubitka informacije o redoslijedu riječi, dok PyRv zadržava visoku točnost od 93% zahvaljujući učinkovitom očuvanju strukture rečenice. U složenijem zadatku označavanja sintaktičkih odnosa, gdje pojedinačni vektori postižu točnost od 71%, prosjek 5 riječi dovodi do naglog pada na 34%. S druge strane, PyRv kompozicija podiže točnost na 77%, što potvrđuje njegovu sposobnost integracije konteksta u koherentne i informativne reprezentacije. Time se dokazuje učinkovitost kompozicionalnog pristupa koji PyRv omogućuje u stvarnim NLP zadacima.

Glavni doprinos ovog rada je predstavljanje metode za kompoziciju višerječnih izraza u objedinjene vektorske prikaze pomoću PyRv-a. Prikazali smo evaluaciju njegove učinkovitosti u usporedbi s metodom prosjeka vektora riječi i potvrdili njegovo kompozicionalno svojstvo. Također smo analizirali strukturu reprezentacijskog prostora koju PyRv generira. Trenirajući model na hrvatskim tekstovima, pokazali smo njegovu prilagodljivost i potencijalnu primjenu na raznim jezicima.

U budućem radu mogla bi se dalje istraživati učinkovitost metode PyRv kao kompozicijskog modela. Prirodan sljedeći korak bila bi njezina evaluacija na standardnim engleskim benchmarkovima za zadatke koji su izrazito osjetljivi na semantičku kompoziciju. To bi također omogućilo izravnu usporedbu s drugim kompozicijskim strategijama, posebice onima koje rabe najsuvremeniji modeli temeljeni na transformerima. Istraživanje arhitektonskih varijacija, poput obogaćivanja rekurzivne funkcije mehanizmima pažnje, također predstavlja obećavajući pravac istraživanja.

Zahvala

Ovaj rad je u potpunosti financiran iz projekta Sveučilišta u Rijeci uniri-iskusni-drustv-23-95.

References / Literatura

Arora, S., Liang, Y., & Ma, T. (2017). A simple but tough-to-beat baseline for sentence embeddings. *International conference on learning representations*.

Babić, K., Guerra, F., Martinčić-Ipšić, S., & Meštrović, A. (2020). A comparison of approaches for measuring the semantic similarity of short texts based on word embeddings. *Journal of information and organizational sciences*, *44*(2), 231–246.

Babić, K., & Meštrović, A. (2024). Recursively autoregressive autoencoder for pyramidal text representation. *IEEE access*, *12*, 71361–71370.

Bahdanau, D. (2014). Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, *5*, 135–146.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, *33*, 1877–1901.

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*.

Grave, E., Bojanowski, P., Gupta, P., Joulin, A., & Mikolov, T. (2018). Learning word vectors for 157 languages. *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Joulin, A., Grave, E., Bojanowski, P., & Mikolov, T. (2016). Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Ljubešić, N., Klubička, F., Agić, Ž., & Jazbec, I.-P. (2016). New inflectional lexicons and training corpora for improved morphosyntactic annotation of croatian and serbian. *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, 4264–4270.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *arXiv:1301.3781*.

Pennington, J., Socher, R., & Manning, C. D. (2014). Glove: Global vectors for word representation. *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Socher, R., Perelygin, A., Wu, J., Chuang, J., Manning, C. D., Ng, A. Y., & Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631–1642.

Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

Wieting, J., Bansal, M., Gimpel, K., & Livescu, K. (2015). Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*.

Zhao, H., Lu, Z., & Poupart, P. (2015). Self-adaptive hierarchical sentence model. *arXiv preprint arXiv:1504.05070*.

A Appendix / Dodatak

Table 5. DEPREL evaluation results using the fastText embedding method (single-word embeddings), where P stands for Precision, R for Recall, F1 for F1 score, and S for Support.
Tablica 5. Rezultati DEPREL evaluacije korištenjem fastText metode reprezentiranja (jednorječne vektorske reprezentacije), pri čemu P označava preciznost, R odziv, F1 F1-mjeru, a S potporu.

Class	P	R	F1	S
punct	1	1	1	3037
nmod	0.6	0.58	0.59	2437
case	0.96	0.98	0.97	2364
amod	0.79	0.93	0.85	2355
nsubj	0.53	0.66	0.59	1725
obl	0.48	0.57	0.52	1607
root	0.45	0.67	0.53	1136
conj	0.19	0.02	0.03	1134
obj	0.49	0.44	0.46	1072
aux	0.75	0.96	0.84	1037
cc	0.83	0.97	0.89	887
advmod	0.8	0.88	0.84	825
flat	0.54	0.69	0.61	689
mark	0.85	0.91	0.88	471
acl	0.44	0.08	0.14	452
cop	0.58	0.22	0.32	415
det	0.87	0.87	0.87	400
xcomp	0.61	0.7	0.65	350
expl	0.85	1	0.92	302
parataxis	0.63	0.38	0.47	300
ccomp	0.18	0.02	0.03	230
discourse	0.48	0.21	0.29	208
advcl	0.29	0.08	0.12	198
nummod:gov	0.72	0.9	0.8	187
appos	1	0.01	0.02	130
nummod	0.82	0.63	0.71	117
fixed	0.34	0.33	0.34	100
csubj	0	0	0	40
det:numgov	0.73	0.66	0.69	29
orphan	0	0	0	13
advmod:emph	0	0	0	5
flat:foreign	0	0	0	4
vocative	0	0	0	3
compound	0	0	0	1
macro avg	0.52	0.48	0.47	
weighted avg	0.68	0.71	0.68	

Table 6. DEPREL evaluation results using the fastText embedding method (averaging embeddings of five words), where P stands for Precision, R for Recall, F1 for F1 score, and S for Support.
Tablica 6. Rezultati DEPREL evaluacije korištenjem fastText metode reprezentiranja (prosjek vektorskih reprezentacija pet riječi), pri čemu P označava preciznost, R odziv, F1 F1-mjeru, a S potporu.

Class	P	R	F1	S
punct	0.54	0.71	0.61	3037
nmod	0.42	0.35	0.39	2437
case	0.28	0.56	0.37	2364
amod	0.3	0.4	0.34	2355
nsubj	0.3	0.3	0.3	1725
obl	0.29	0.03	0.06	1607
root	0.26	0.17	0.21	1136
conj	0.27	0.38	0.32	1134
obj	0.26	0.12	0.16	1072
aux	0.35	0.34	0.35	1037
cc	0.22	0.25	0.24	887
advmod	0.29	0.24	0.26	825
flat	0.44	0.43	0.44	689
mark	0.33	0.27	0.3	471
acl	0.23	0.07	0.11	452
cop	0.22	0.14	0.17	415
det	0.25	0.13	0.17	400
xcomp	0.35	0.15	0.21	350
expl	0.26	0.35	0.3	302
parataxis	0.76	0.09	0.16	300
ccomp	0	0	0	230
discourse	0.67	0.03	0.06	208
advcl	0.14	0.01	0.01	198
nummod:gov	0.27	0.6	0.37	187
appos	0.2	0.02	0.04	130
nummod	0.24	0.13	0.17	117
fixed	0.31	0.24	0.27	100
csubj	0	0	0	40
det:numgov	0	0	0	29
orphan	0	0	0	13
advmod:emph	0	0	0	5
flat:foreign	0	0	0	4
vocative	0	0	0	3
compound	0	0	0	1
macro avg	0.25	0.19	0.19	
weighted avg	0.34	0.34	0.31	

Table 7. DEPREL evaluation results using the PyRv+FT embedding method (composing embeddings of five words), where P stands for Precision, R for Recall, F1 for F1 score, and S for Support.
Tablica 7. Rezultati DEPREL evaluacije korištenjem PyRv+FT metode reprezentiranja (kompozicija vektorskih reprezentacija pet riječi), pri čemu P označava preciznost, R odziv, F1 F1-mjeru, a S potporu.

Class	**P**	**R**	**F1**	**S**
punct	1	1	1	3037
nmod	0.74	0.71	0.73	2437
case	0.98	0.97	0.97	2364
amod	0.81	0.82	0.82	2355
nsubj	0.7	0.7	0.7	1725
obl	0.59	0.63	0.61	1607
root	0.61	0.69	0.65	1136
conj	0.66	0.62	0.64	1134
obj	0.52	0.65	0.58	1072
aux	0.78	0.94	0.85	1037
cc	0.91	0.95	0.93	887
advmod	0.68	0.82	0.74	825
flat	0.77	0.69	0.73	689
mark	0.91	0.91	0.91	471
acl	0.6	0.47	0.53	452
cop	0.69	0.39	0.5	415
det	0.72	0.69	0.71	400
xcomp	0.67	0.59	0.63	350
expl	0.86	0.99	0.92	302
parataxis	0.83	0.56	0.67	300
ccomp	0.41	0.16	0.23	230
discourse	0.65	0.52	0.58	208
advcl	0.38	0.22	0.28	198
nummod:gov	0.82	0.76	0.79	187
appos	0.42	0.38	0.4	130
nummod	0.67	0.76	0.71	117
fixed	0.67	0.61	0.64	100
csubj	0	0	0	40
det:numgov	0.5	0.52	0.51	29
orphan	0	0	0	13
advmod:emph	0	0	0	5
flat:foreign	0	0	0	4
vocative	0	0	0	3
compound	0	0	0	1
macro avg	0.58	0.55	0.56	
weighted avg	0.77	0.77	0.76	

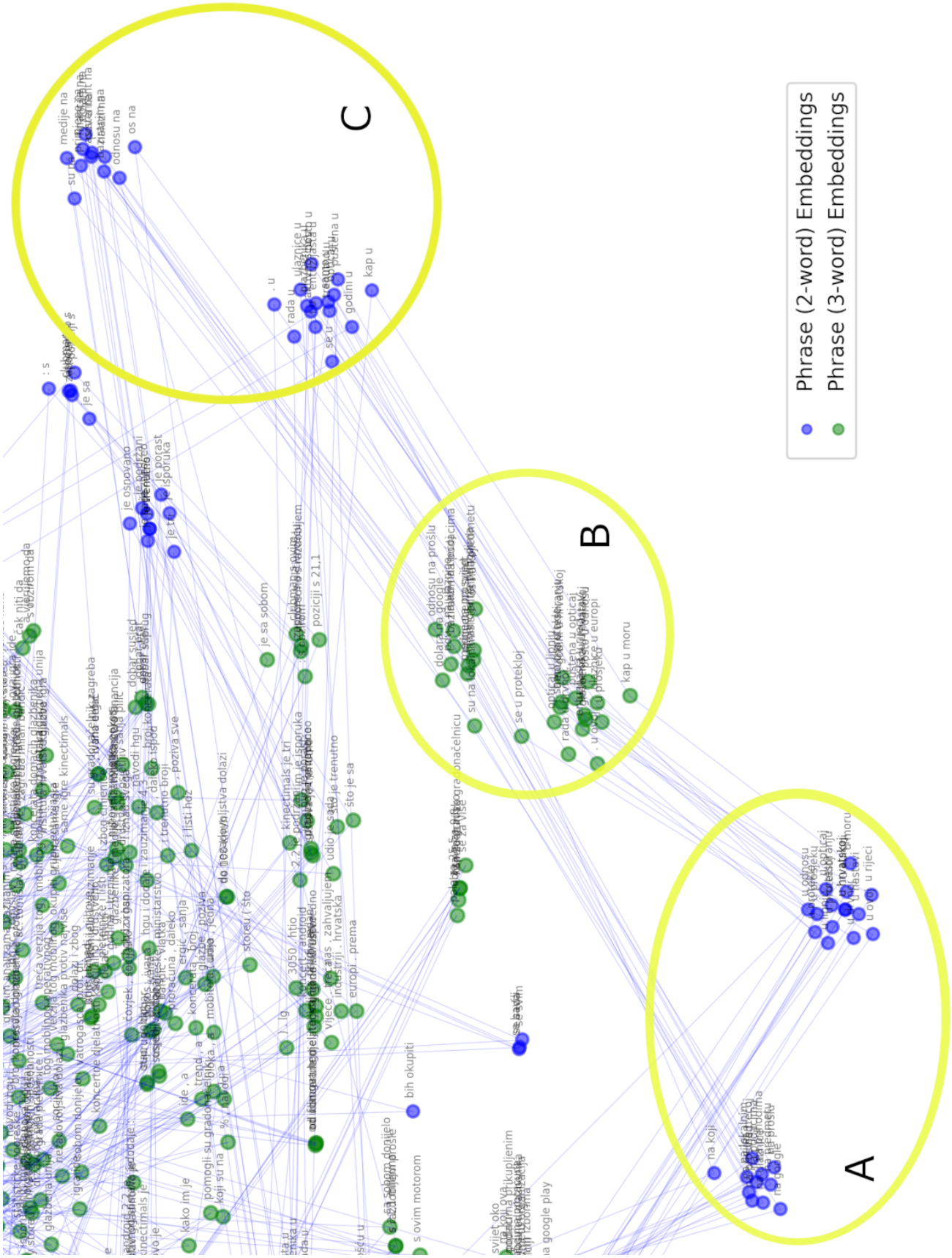


Figure 4: Visualization of the PyRv+FT representation space (reduced from 300 dimensions using t-SNE). Highlighted areas A, B, and C contain phrases with the prepositions 'na' (eng. 'on') and 'u' (eng. 'in'). Tables 3 and 4 (translated: 3 and 4) provide detailed examples of these phrases and their connections within the space.
Slika 4: Vizualizacija PyRv+FT reprezentacijskog prostora (reduciranog s 300 dimenzija korištenjem t-SNE-a). Istaknuta područja A, B i C sadrže fraze s prijedlozima 'na' (engl. 'on') i 'u' (engl. 'in'). Tablice 3 i 4 (prevedeno: 3 i 4) donose detaljne primjere ovih fraza i njihovih veza unutar prostora.

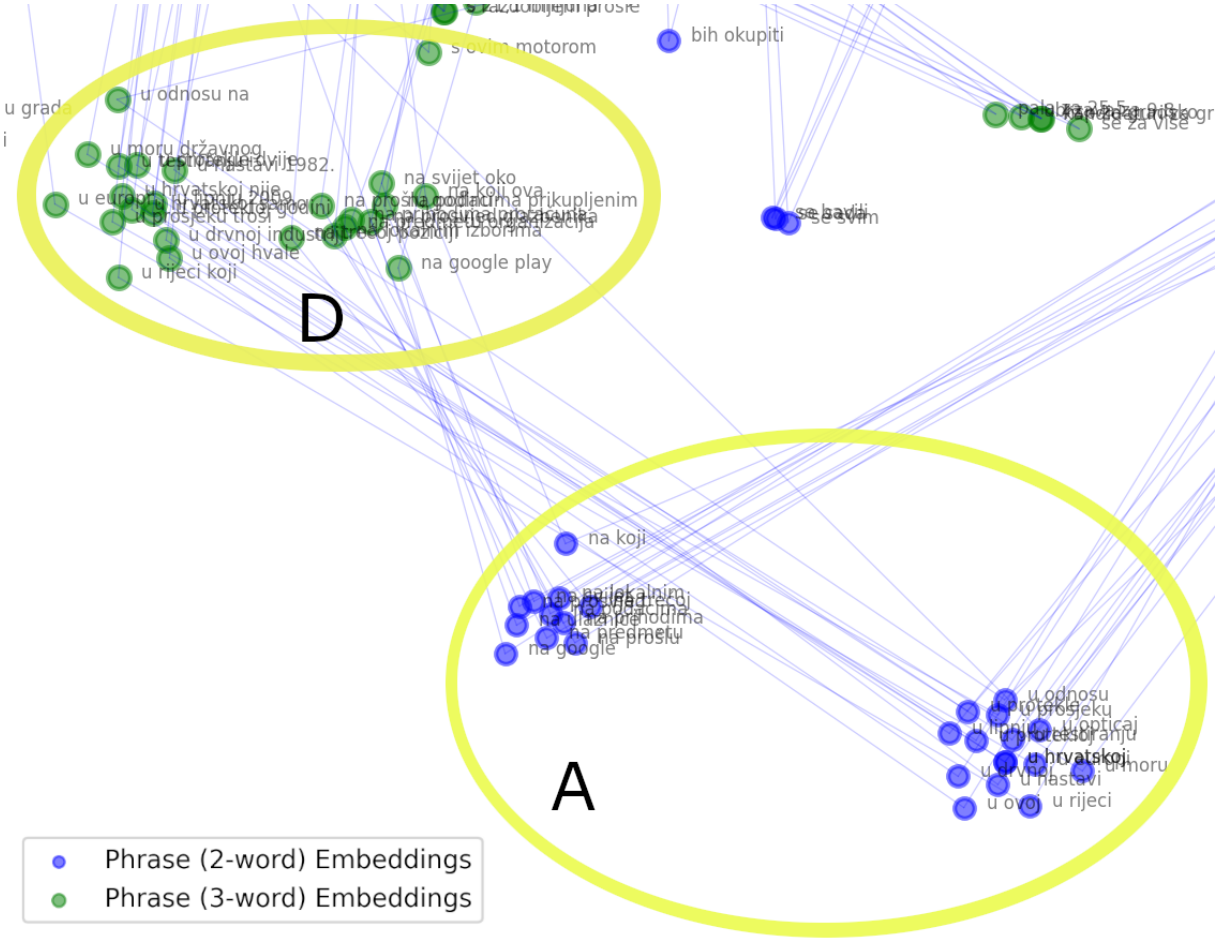


Figure 5. Visualization of the PyRv+FT representation space (reduced from 300 dimensions using t-SNE). Highlighted areas A and D contain phrases structured around the prepositions 'na' (eng. 'on') and 'u' (eng. 'in'). Tables 3 and 4 (translated: 3 and 4) present examples of these phrases and their relationships in the embedding space.
Slika 5. Vizualizacija PyRv+FT reprezentacijskog prostora (reduciranog s 300 dimenzija korištenjem t-SNE-a). Istaknuta područja A i D sadrže fraze strukturirane oko prijedloga 'na' (engl. 'on') i 'u' (engl. 'in'). Tablice 3 i 4 (prevedeno: 3 i 4) donose primjere ovih fraza i njihovih odnosa u vektorskom prostoru.