machine learning & knowledge extraction

MDPI

Article

Unsupervised Word Sense Disambiguation Using Transformer's Attention Mechanism

Radu Ion, Vasile Păiș, Verginica Barbu Mititelu, Elena Irimia, Maria Mitrofan, Valentin Badea and Dan Tufiș *

Research Institute for Artificial Intelligence "Mihai Drăgănescu", "Calea 13 Septembrie", 050711 Bucharest, Romania; radu@racai.ro (R.I.); vasile@racai.ro (V.P.); vergi@racai.ro (V.B.M.); elena@racai.ro (E.I.); maria@racai.ro (M.M.); valentin.badea@racai.ro (V.B.)
* Correspondence: tufis@racai.ro

Abstract: Transformer models produce advanced text representations that have been used to break through the hard challenge of natural language understanding. Using the Transformer's attention mechanism, which acts as a language learning memory, trained on tens of billions of words, a word sense disambiguation (WSD) algorithm can now construct a more faithful vectorial representation of the context of a word to be disambiguated. Working with a set of 34 lemmas of nouns, verbs, adjectives and adverbs selected from the National Reference Corpus of Romanian (CoRoLa), we show that using BERT's attention heads at all hidden layers, we can devise contextual vectors of the target lemma that produce better clusters of lemma's senses than the ones obtained with standard BERT embeddings. If we automatically translate the Romanian example sentences of the target lemma into English, we show that we can reliably infer the number of senses with which the target lemma appears in the CoRoLa. We also describe an unsupervised WSD algorithm that, using a Romanian BERT model and a few example sentences of the target lemma's senses, can label the Romanian induced sense clusters with the appropriate sense labels, with an average accuracy of 64%.

Keywords: unsupervised word sense disambiguation; word sense induction; k-means clustering; Transformer; BERT; attention mechanism; CoRoLa; Romanian; English

Academic Editor: Laura Po

Received: 25 October 2024
Revised: 9 January 2025
Accepted: 15 January 2025
Published: 18 January 2025

Citation: Ion, R.; Păiș, V.; Mititelu, V.B.; Irimia, E.; Mitrofan, M.; Badea, V.; Tufiș, D. Unsupervised Word Sense Disambiguation Using Transformer's Attention Mechanism. *Mach. Learn. Knowl. Extr.* **2025**, *7*, 10. https://doi.org/10.3390/make7010010

1. Introduction

With the invention of the Transformer model [1], the Natural Language Processing (NLP) field of Artificial Intelligence (AI) received a huge boost, both in popularity and, mainly, in the ability to fulfill its main goal, that of a computer understanding natural language at human competency level. Recently, Large Language Models (LLMs) such as OpenAI's GPT-4 Omni [2] and Google's Gemini 1.5 [3] have been developed that can take on tasks previously unapproachable for computers, such as valid computer code generation, solving challenging math problems, domain-independent and highly specialized question answering from, e.g., physics or chemistry, and so on. With such a rapid advancement of the NLP field, one can question the motives behind further studying subfields of NLP that were thought to be the building blocks of a computer's understanding of natural language, e.g., dependency parsing or word sense disambiguation. Yet, one fact remains about the rapid progress of the NLP field, achieved using LLMs: there is no scientific explanation of the language-understanding capabilities of LLMs, other than the sheer number of parameters (topping one trillion for GPT-4o and Gemini 1.5) that can "remember" every piece of text that anyone has ever written.

Word sense disambiguation (WSD) is the NLP task of automatically determining the sense (identifier) with which a target word (a word we wish to disambiguate) appears in a sentence. The sense inventory of the target word is predetermined and contains a list of senses that the word can have (at a very minimum, the sense inventory contains the sense identifier and the textual definition of each sense). Word sense induction (WSI) refers to the task of automatically clustering the occurrences of a target word into sets in which occurrences *have the same sense*. Thus, an *unsupervised* WSD algorithm can perform WSI on occurrences of the target word in the chosen sentence sample, followed by automatic mapping of a sense identifier to each cluster [4]. In contrast, a *supervised* WSD *algorithm* can directly find the sense identifier of the target word in a sentence, provided it was previously trained on sentences in which the target word was annotated with the appropriate sense in the context.

Transformer models, once pretrained on very large corpora, offer access to their attention tensors, stored at each hidden level. Simply put, the attention matrix of a sentence stores a weight between 0 and 1 of the "contextual relevance" of the word at position *i* with the word at position *j* in the tokenized sentence. The contextual relevance can be explained at different linguistic levels, from the morphological level to the syntactic and semantic levels. For instance, in Romanian, we can observe a high attention weight between the singular form of a masculine noun and its enclitic article, tokenized away by the WordPiece tokenizer [5], as is "băiat ##ul" (boy ##the$_{\text{masc. sg.}}$).

Our Contribution

With respect to the WSI task, the attention mechanism of the Transformer model offers direct and quantitative insight into what words are semantically related to the target word. The typical BERT model [6] contains between 12 and 24 hidden layers which are pipelined to offer the final contextualized embedding for the target word, but each hidden layer contains 8 to 16 attention heads, each with its own parameters. We see that, by mining the attention matrix at each hidden layer and each attention head, we can easily build highly dimensional, attention-based contextual vectors for our target word, and this is one of the main contributions of this paper: we show that by concatenating the BERT-generated contextual embedding of the target word with the attention-based contextual vector we introduce here, we obtain better clustering results of word senses.

The second contribution of the paper is leveraging another language for WSI, namely English in our case, to help automatically determine the number of senses that a Romanian target word has in our corpora. We base our current work on the same principle [7] we used when we developed the Romanian WordNet [8], aligned to the Princeton WordNet [9] at the synset level: the translation preserves meaning. Thus, we assume that the number of senses of the target word in a sample of Romanian sentences is the same as (or close to) the number of senses of the various translation equivalents of the target word in English, in the automatically translated to English sentence sample. By enforcing this constraint to the clusters of target word's senses in Romanian and English and by measuring the Romanian and English clusters' overlap, we can determine the number of clusters in Romanian and in English that maximize the overlap measure.

The third contribution of this paper is an unsupervised WSD algorithm that, given the clusters of word senses, learns a sense mapping from textual sense definitions to occurrences of the target word in a cluster, using BERT models. The unsupervised WSD algorithm only needs a tiny amount of example sentences (currently five) for each sense of the target word, positioning the learning of the sense definition to cluster mapping task as a few-shot learning task [10].

In the rest of this paper, we review related work in Section 2, we describe the lexical sample of Romanian lemmas with which we work in Section 3 and we provide detailed descriptions of the WSI and WSD algorithms we introduce in this paper in Section 4. Section 4 also contains a case study of the disambiguation of the adjective "aerian" (aerial), including parameter optimization results. Section 5 discusses the results of the optimal WSD algorithm on our selected Romanian lemma sample and Section 6 concludes the paper.

2. Related Work

Supervised WSD is essentially a classification task in which a classifier is trained on a corpus in which the target word is manually annotated with the appropriate sense and then is run on a fresh text, picking the most probable sense of the target word, according to the learned model. On the other hand, unsupervised WSD is essentially a clustering task, in which many examples of the target word are clustered together, using features that pertain to the sense of the word. After the clusters are created, one can develop a method of automatically mapping the examples of each cluster to a sense of a target word from the sense inventory, thus achieving the same goal of supervised WSD, namely, the annotation of each occurrence of the target word (in the cluster) with the mapped sense ID. This is the general strategy we adopt in this paper.

Even though supervised WSD is (still) superior to unsupervised WSD [11], manually creating sense-annotated corpora for many words and many of their occurrences, especially when the sense distribution of words is highly skewed towards a frequent sense (this is why the most frequent sense for sense inventories that provide this information is the baseline algorithm for any WSD algorithm), is unachievable for any practical use of WSD. Thus, the complementary domain of WSI [12,13] has emerged to deal with unsupervised (i.e., working with corpora that are not annotated with senses) clustering of word senses. The current, main research directions in WSI are as follows:

1. Non-Transformer-based word sense embeddings [14–16];
2. Topic models for word sense embeddings [17];
3. Transformer-based word sense embeddings [18,19].

Since we also use a BERT model to extract word sense embeddings, we describe the latter papers and compare their approach to ours.

The core idea of Eyal et al. [18] is that word senses are well described by contextualized embeddings produced by BERT models but also by vectors that contain very probable words that can replace the target word, with the same or related meaning, as features. For example, if we would want to distinguish between the "insect" sense of the noun "bug" and its "disease" sense, we would invoke the BERT model to suggest the most probable five words to replace "bug", and according to the cited paper, these replacement words would be the following:

- "insect", "fly", "beetle", "bugs" and "worm" for the "insect" sense;
- "virus", "infection", "crisis", "disease" and "surprise" for the "disease" sense.

By constructing a vocabulary of possible word replacements for all occurrences of the target word, one can build a real-valued vector for each occurrence of the target word. The result obtained by this system in the SemEval-2010 Task 14 of WSI has a V-Measure [12] of 40.7%, a spectacular increase from the best system at that competition that obtained only 16.2% [12]. Compared to our WSI algorithm, Eyal et al. use word substitutes for the target word to create contextual embeddings, while we use BERT's attention layer weights to find words cooccurring with the target word that are indicative of its sense.

PolyLM [19] is an unsupervised sense embedding model, wishing to solve the problem of multiple senses of a word being collapsed into a single contextual embedding provided

by a BERT model. This sense conflation problem is resolved by PolyLM by learning a probability distribution over the words and their senses at a masked position [6] in the sentence, assuming that the probability of a word occurring at a certain position is the sum of the word's sense probabilities in the training corpus, and that a sense is much more probable in some contexts than the rest of the senses. PolyLM obtained a V-Measure of 40.5% in the SemEval-2010 Task 14 of WSI, thus being on par with the word substitution algorithm presented above. With respect to our WSI algorithm, PolyLM models construct word sense embeddings directly, while we construct sense embeddings from BERT's attention layers. Furthermore, PolyLM is not tested in the WSD task.

The Transformer-based supervised WSD baseline is described by Chawla et al. [20], who round up nine of the widely adopted Transformer models and try them on the WSD task at SensEval-2 and SensEval-3 [11]. The WSD algorithm is straightforward: for each test word, obtain its contextualized embedding from the selected Transformer model and carry out a k-nearest neighbor search for its most k similar embeddings in the training (sense-annotated) set. The most frequent sense label from the most similar k-annotated examples wins. Chawla et al. [20] find that this method is very good, reporting an F-1 measure of 76.81% in SensEval-2 and 80.96% in SensEval-3, both with the BERT model.

Vandenbussche et al. [21] fine-tune a BERT model for the supervised WSD task. They solve the binary classification problem that when an algorithm is given a target word in an input sentence, coupled with a definition of one of its senses, the algorithm has to say if the word is/is not used with the sense described by the definition. They find that joining the input sentence and the sense definition with the "[SEP]" keyword and taking the average of the output embeddings of the result onto which they stack a fully connected, two-neuron layer for binary classification yields the best results. BERT is, again, the best model for the job, achieving an F-1 measure of 76.2%.

Both Chawla et al. [20] and Vandenbussche et al. [21] fine-tune BERT models for the supervised WSD classification task. In contrast, we perform few-shot learning of mappings from sense definitions to clusters, on Romanian clusters of example sentences of the target word built with attention-based contextual vectors. Thus, our method is essentially an unsupervised (or weakly supervised) WSD approach which does not rely on sense-annotated corpora.

Tripodi and Navigli [22] take a game theoretic approach to knowledge-based WSD. In short, a whole text is disambiguated at once by a set of "players" which are the content words of the text (nouns, verbs, adverbs and adjectives). Each player can choose from a set of "strategies", which are the possible senses extracted from a lexical knowledge base, and the whole system computes a payoff matrix which depends on a word similarity matrix A and a sense similarity matrix Z. This knowledge-based WSD system (which is essentially an unsupervised WSD system that uses a structured sense inventory, such as WordNet, or has access to some sense-related information without explicitly training on sense-annotated corpora) produces a 67.7% F-1 measure on the disambiguation task on all available WSD test sets, which is close to the 71.5% achieved by the best supervised WSD system. This is the only method we are aware of that uses the attention mechanism of the BERT models to provide semantic cues to word pairs in a sentence, but it only uses the attention at the last layer. Our contextual vectors are constructed using all attention layers of the BERT model, and the optimal contextual vectors are determined automatically by hyperparameter tuning.

3. Lexical Sample Selection

To experiment with our WSI and WSD algorithms, we need a set of content word lemmas that have the following qualities:

1. Have at least two senses in the Romanian Explanatory Dictionary (DEX) [23], but no more than five, to ease manual gold-standard annotation for each lemma. We strived to include words whose senses are rather close to each other, as well as words with senses that are more semantically distant. The evaluation of the distance between senses was based on the linguist's intuition that made the choices and who is also in charge with the development of the Romanian WordNet.
2. Are not homonymous. Unlike polysemy, homonyms are very distant semantically, and presumably, this makes the tasks of semantic disambiguation rather easy, given that the contexts of occurrence of one homonym are totally different from those of the other homonym.
3. Do not have rich collocation-driven senses. Whenever the senses described in the dictionary are exemplified with collocates, we leave the respective word aside, as collocates help to easily figure out the meaning (be it manually or automatically). The same applies to words whose senses are expression-dependent.
4. Do not appear in the Romanian WordNet, as we are also interested in automatically extending the Romanian WordNet with new synsets (work outside the scope of this paper).
5. Are frequent in the Reference Corpus of the Contemporary Romanian Language (CoRoLa) [24], as our unsupervised WSD algorithm needs a relatively large sample of sentences to train itself to map prototype sense examples to sense clusters. This criterion also ensures that selected lemmas have high coverage in other corpora.

Table 1 contains the first 10 most frequent lemmas of nouns, verbs, adjectives and adverbs from the CoRoLa, while Table 2 contains the selected sets of lemmas we worked with. For each lemma, we randomly shuffle all its example sentences and select 5000 examples as our working set. For lemmas which have fewer than 5000 examples, we take all of them. All example sentences of our lexical sample are processed with the Romanian-specialized Rodna text processor [25], a sentence-splitting, tokenization, POS-tagging, lemmatization and dependency-parsing pipeline targeting high-performance Romanian text processing.

Table 1. Ten most frequent word lemmas and their frequency from the CoRoLa (the English translation is of the most frequent sense of the lemma).

Nouns		**Verbs**		**Adjectives**		**Adverbs**	
articol (article)	2,671,344	putea (can)	2,659,565	dat (given)	1,810,237	poate (maybe)	1,197,761
caz (case)	1,828,842	publica (publish)	2,606,670	prevăzut (provided)	1,623,604	numai (only)	561,452
lege (law)	1,706,469	prevedea (foresee)	1,714,764	public (public)	1,487,763	astfel (thus)	474,963
dată (date)	1,637,314	face (do)	1,668,716	oficial (official)	1,251,986	doar (just)	341,220
an (year)	1,599,338	trebui (must)	1,199,171	publicat (published)	1,186,120	bine (well)	277,278
parte (part)	1,397,337	sta (stay)	902,738	național (national)	1,070,671	acum (now)	224,377
persoană (person)	1,362,166	stabili (establish)	811,111	prezent (present)	1,033,534	apoi (then)	222,453
stat (state)	1,305,742	da (give)	780,371	următor (next)	901,561	așa (so)	208,226
activitate (activity)	1,200,049	avea (have)	726,615	mare (big)	824,501	încă (yet)	190,663
serviciu (job)	1,102,383	modifica (modify)	701,571	medical (medical)	783,207	aici (here)	181,573

Table 2. Content word lemmas for the lexical sample (the English translation is of the most frequent sense of the lemma).

Nouns		Verbs		Adjectives		Adverbs	
pondere (weight)	24,398	abilita (authorize)	31,372	oficial (official)	1,251,986	aci (here)	4481
caiet (notebook)	22,203	disputa (play)	12,366	unic (unique)	273,681	orbește (blindly)	1201
incintă (premise)	18,329	dispera (despair)	8111	cult (cultivated)	58,432	zdravăn (healthy)	1068
relief (terrain)	8377	recepționa (receive)	6900	aerian (aerial)	54,152	omenește (humanly)	831
codru (forest)	7027	răci (cool)	6135	conform (consistent)	34,861		
puț (well)	4804	depărta (separate)	6060	reprezentativ (representative)	27,291		
papuc (slipper)	2900	gripa (grind to a halt)	2847	verbal (verbal)	26,467		
brumă (frost)	2558	înseta (long for water)	2436	vegetal (vegetable)	25,295		
ansă (loop)	1536	parveni (become rich)	2010	sectorial (sectorial)	23,094		
săpuneală (scolding)	56	înfrăți (bond)	989	liric (lyrical)	17,960		

Given the high polysemy of nouns and verbs (which directly correlates with the frequency, i.e., a polysemous word will occur more frequently than a monosemous one), as well as the fact that most polysemous words in the Romanian WordNet are nouns (more than 66% of the total number of literals [8]), the frequencies listed in Tables 1 and 2 are not surprising: nouns and verbs are more polysemous than adjectives, which, in turn, are more polysemous than adverbs. Thus, we had difficulties selecting 10 adverbs that are polysemous; we found 4 that have only two senses each.

To evaluate our WSI and WSD algorithms, we need a small gold standard of sense annotations for each lemma in Table 2. Thus, for each lemma, from the random sample of 5000 (or less; see Table 2) examples, we randomly selected 200 examples to be manually annotated. Each lemma is presented within a sentence-long context, and the annotator's task is to read the whole sentence and pick the right sense from the respective lemma's set of senses extracted from DEX. We further present some problematic annotation cases.

Case A: Difficulty assigning a sense. Rarely does it happen that the sentence-long context is not enough for assigning a sense to the target word. But it can be the case that the other content words in the respective sentence are not helpful for deciding upon the sense of the target word. In ex. (1), the presence of the word "ferestrele" (windows) in a part–whole syntactic structure (the genitive construction) makes the meaning of the target word "puțului" (shaft + [of the]$_{\text{gen.,\,sg.,\,masc.}}$; lemma "puț") unclear because the reader understands that the "puț" part indicates the presence of windows, which is against the common knowledge about it. Thus, another meaning may be necessary.

> De la toate ferestrele **puțului** locatarii zbierau la ei, dar jos, în gang, era pustiu, nici urmă de copii. (1)

> From all the windows **of the shaft**, the tenants shouted at them, but down in the gang, it was deserted, no sign of children. (2)

Example (3) below probably originates in imaginative writing, where word combinations are sometimes striking or figurative (poetic example):

C-am auzit bietul pat/Te striga, să nu **răcească**. (3)

I heard the poor bed/Calling you, afraid not to **get cold**. (4)

Sometimes, in poetry, due to constraints on the number of syllables in a line, clitics may be left out. In example (3), the meaning "to get cold" of the verb "răci" is lexicalized in the absence of the reflexive clitic "se" (which is actually specific to this meaning), which may be misleading in interpretation.

Case B: Impossibility to assign only one sense. The context may not contain any word that could help distinguish between two senses of a target word: in ex. (5), either of the two senses of "puțul" (well + the$_{\text{nom.,\,sg.,\,masc.}}$; lemma "puț") can be instantiated; thus, both were manually assigned.

Habar n-aveau cât de adânc era **puțul**. (5)

They had no idea of how deep **the well/shaft** was. (6)

The sense of "well" and the sense of "shaft" of the word "puțul" are equally probable in this sentence, as the English translation in (6) shows.

Case C: Necessity to coin a new word sense. As generally admitted, dictionaries are incomplete works due to incomplete coverage either of the lexical inventory of a language or of the whole inventory of senses of some of the words they describe. Whenever no sense from the set of senses already defined for a target word in DEX applies to a context, it is marked as requiring an additional sense.

Ce, nu ești **zdravăn**? (7)

What is the matter, are you not **mentally healthy**? (8)

The meaning "mentally healthy", corresponding to the word "zdravăn" in example (7), is not recorded in DEX and should be added.

Case D: Impossibility of assigning a meaning due to the incorrect tokenization and/or POS tagging or due to missing diacritics. In Romanian, omitting diacritical marks (i.e., ș, ț, ă, î and â) may introduce ambiguities, e.g., compare "fata" (the girl) with "față" (a face). Example (9) shows a case where the target word "cultului" (lemma "cult") is POS-tagged incorrectly as an adjective instead of a noun. This is a very common POS-tagging mistake of Romanian POS taggers, as nouns and adjectives have the same inflections and appear in similar contexts.

> În cazul universităților confesionale, organizarea senatului universitar se va face cu respectarea statutului și specificului dogmatic și canonic al **cultului** fondator. (9)

> In the case of confessional universities, the organization of the university senate will be done in compliance with the statute and the dogmatic and canonical specificity of the founding **cult**. (10)

Table 3 contains statistics of the gold-standard annotations for all 34 lemmas from Table 2. Each row in Table 3 sums to 200, the number of examples that were judged, except for the adjective "aerian", which has 1000 annotated examples, which will serve as a case study for parameter optimization (see Section 4.4).

Table 3. Gold-standard annotation statistics.

Lemmas	Annotated	Cases A + D	Case B	Case C
Nouns				
pondere	199	1	0	0
caiet	200	0	0	0
incintă	192	1	2	5
relief	192	7	1	0
codru	179	21	0	0
puț	165	10	9	16
papuc	145	55	0	0
brumă	163	37	0	0
ansă	170	30	0	0
săpuneală	49	6	1	0
Verbs				
abilita	198	2	0	0
disputa	198	1	0	1
dispera	187	12	0	1
recepționa	200	0	0	0
răci	192	8	0	0
depărta	199	1	0	0
gripa	52	148	0	0
înseta	198	2	0	0
parveni	170	2	26	2
înfrăți	200	0	0	0
Adjectives				
oficial	200	0	0	0
unic	199	0	1	0
cult	94	106	0	0
aerian	996	3	0	1
conform	198	2	0	0
reprezentativ	196	0	4	0
verbal	197	1	1	0
vegetal	200	0	0	0
sectorial	200	0	0	0
liric	197	2	1	0
Adverbs				
aci	175	6	0	19
orbește	45	155	0	0
zdravăn	177	0	2	21
omenește	194	6	0	0
Total	6716	619	48	66

It is also worth noting that, in Table 3, we have three lemmas with a lot of Case A + Case D annotation problems: "gripa", "cult" and "orbește". Most of the annotation problems are of Case D, caused by POS-tagging errors. Thus, "gripa" can also be a definite noun (the flu), "cult" can also be a noun (the cult) and "orbește" can also be a verb (to go blind, third person, singular, present tense). All three different readings change the meaning completely.

Our WSI algorithm uses the English translations of the Romanian example sentences on the assumption that meaning is preserved by translation, and thus, the number of clusters in Romanian and English should be similar (more details below). To automatically translate the entire lexical sample sentence set, we used Mistral-7B, ver. 0.3 [26], and

prompted it with the following request: "Translate the following sentence from Romanian to English: <Romanian sentence>". We marked the targeted word in each example sentence with the character sequence "# <word> #", as in the translation example pair (11) and (12):

Escadrila 18 RAF a trimis două bombardiere Bristol Blenheim în misiune în spațiul **# aerian #** belgian, ambele doborâte de vânătoarea germană. (11)

Squadron 18 RAF sent out two Bristol Blenheim bombers on a mission in the Belgian **# airspace #**, both shot down by German fighter planes. (12)

We can retrieve the target word translation by extracting the phrase that is delimited by the hash character. In the example pair (11) and (12), we see that "aerian" has been translated to "airspace", which is, in fact, the translation of "spațiul aerian", but the intended meaning is still captured in the translation, that of "related to aviation". We note here that Mistral usually preserves the hash characters in the translation, but there are cases in which one or even both characters are missing. In these cases, we select the English word which was found as a translation for the Romanian lemma most frequently, in all example sentences of that lemma. Example pair (13) and (14) shows such a translation:

Au fost observate infecții ale gurii, gâtului și căilor **# aeriene #** superioare la copiii care au primit tratament cu Increlex. (13)

Infections of the mouth, throat, and upper **airways** have been observed in children who received treatment with Increlex. (14)

Translation of "aeriene" was not marked with hash characters in example (14), but "airways" is a frequent translation of "aeriene" in the 5000-example sentence set of the lemma "aerian", and as such, we can select it as the translation equivalent in example (14).

Table 4 lists the most frequent English translations for each lemma in the Romanian lexical sample. Bolded translations pertain to one of the senses defined for that Romanian lemma, while the underlined "translations" were failures to translate those Romanian words.

Table 4. Frequent English translation equivalents for Romanian lemmas.

Lemmas	Translations
Nouns	
pondere	**weight**, **share**, **proportion**, **weighting**, **percentage**, significant, number, great, **rate**, population
caiet	**notebook**, **sheet**, job, file, folder, **task**, **taskbook**, **book**, notepad, questionnaire
incintă	**premise**, **building**, **enclosure**, **compound**, **facility**, **area**, container, fortification, **room**, **chamber**
relief	**relief**, **terrain**, **landscape**, Romanian, **hilly**, character, **mountainous**, **raise**, **hill**, **topography**
codru	**forest**, codru, codrul, codrii, **wood**, codrului, codri, codrilor, **bread**
puț	**well**, **pit**, puț, **shaft**, **hole**, **oil**, number, **water**, **wells**, **tank**
papuc	**slipper**, papuc, **shoe**, **sandal**, papuci, **rubber**, papucii, **boot**
brumă	**frost**, **fog**, snow, **mist**, **veil**, **autumn**, **haze**, winter, **frosty**, **cold**
ansă	ansa, **jejunal**, **loop**, **anastomosis**, annex, diathermic, parallel, **intestinal**, **anus**, year
săpuneală	needle, **soap**, **slap**, **face**, **sapooning**, weekly, umbrella, shovel, razor, **puddle**

Table 4. *Cont.*

Lemmas	Translations
Verbs	
abilita	**ability**, **authorize**, **capable**, **enable**, **able**, **competent**, **empower**, **qualify**, **law**, **skill**
disputa	**dispute**, **play**, **match**, **hold**, **debate**, **contest**, **two**, **final**, **place**
dispera	**desperate**, **desperately**, **despair**, **despairingly**, **disappointed**, **despairing**, **disappoint**, **sadly**
recepționa	**receive**, **reception**, **accept**, refer, separately, **message**, **information**, **electronic**, works
răci	**cool**, **cold**, down, **get**, **freeze**, **cooling**, **chill**, **temperature**, **refrigerate**
depărta	**away**, **depart**, **leave**, **remove**, **far**, **distance**, **distant**, **withdraw**, **apart**, **separate**
gripa	**sick**, gripat, **engine**, **gripe**, **grind to a halt**, **falter**, economic, word, **stir**, **stiff**
înseta	**hungry**, **craving**, **starve**, **eager**, **insatiable**, **yearn**, **long**, enchant, **thirsty**, **famish**
parveni	**reach**, come, **arrive**, **receive**, **succeed**, parvin, **become**, **climb**, person
înfrăți	**twin**, **friend**, **connect**, **link**, **fraternize**, **bond**, **unite**, **city**, **together**
Adjectives	
oficial	**official**, oficial, august, **officially**, **article**, translate, **publish**
unic	**unique**, unic, number, publish, **only**, use, payment
cult	**cult**, **culture**, **faith**, **Christian**, **cultured**, **religious**, **worship**, **pious**, **church**, **religion**
aerian	**air**, **aerial**, **aviation**, **airline**, **airspace**, aerian, **airway**, **airborne**, aeriene
conform	**accordance**, **conform**, **conformity**, **accord**, copy, line, **compliance**, council, **conforming**
reprezentativ	**representative**, **representation**, **represent**, **representatively**, **renown**, **representational**, **national**, **team**, Romanian
verbal	**verbal**, **verbally**, **verb**, reception, **orally**, **process**, **procedural**, **note**, minute, **write**
vegetal	**vegetable**, **vegetal**, **plant**, **vegetation**, **animal**, **product**, **origin**, **production**, **agricultural**, **vegetarian**
sectorial	**sectorial**, **sectoral**, **sectorially**, **sector**, **development**, pension, **economic**, **strategy**, **operational**, **program**
liric	**lyrical**, **lyric**, **literary**, **poetry**, **poetic**, **lyricism**, **lyrically**, **literature**, **lyricist**, **poet**, **poem**
Adverbs	
aci	**here**, aci, there, one, place, only
orbește	orbește, **blindly**, shamelessly
zdravăn	**healthy**, zdravăn, **healthily**, **health**, **sick**
omenește	**human**, **person**, **humanly**, **humanity**, **humanely**, **humanize**, **humane**, **man**, **humankind**, **humanizing**

4. WSI and WSD Algorithms

As already mentioned, our unsupervised WSD algorithm uses the following general strategy:

1. Cluster example sentences (i.e., perform WSI) of the target lemma L in both Romanian and English such that the number of clusters in both languages, obtained independently, optimizes a cluster overlap measure. We adopt the V-Measure cluster overlap measure, the one used in SemEval-2010 Task 14 [12]. Intuitively, if translation conserves the meaning, we expect to obtain roughly the same clusters in Romanian and English, such that a cluster overlap measure would have an optimum value for a similar number of clusters in Romanian and English.
2. In Romanian, train a BERT WSD model on each cluster c_j of L to estimate the probability $P(s_i, c_j)$ of assigning sense s_i of lemma L to cluster c_j. Together with the probability of cluster c_j, $P(c_j)$, computed from the classes' distribution on occurrences of lemma L, maximize the conditional probability $P(s_i|c_j) = \frac{P(s_i,c_j)}{P(c_j)}$.

4.1. The WSI Algorithm

The clustering algorithm works with any of the following types of contextual vectors describing the target lemma in an example sentence:

1. A BERT contextual vector: Given lemma L, search for its occurrence in the BERT-tokenized example sentence and take the BERT embedding from the last hidden state. If the occurrence of lemma L is split by the BERT tokenizer, take the element-wise sum of the BERT embeddings of the sub-tokens.
2. An attention contextual vector: See below for a detailed description.
3. A concatenation of the two: Only concatenate the BERT contextual vector with the attention contextual vector.

The attention contextual vector of a lemma L is a vector whose dimensions are indexed by lemmas that were close, in terms of attention weights, to the target lemma L across all the example sentences. All these lemmas form a vocabulary $\boldsymbol{V}$, which is a list of alphabetically sorted lemmas. If lemma l appears in the attention matrix of an example sentence of L at some hidden level and attention head, then the attention weight between L and l is going to count (we explain below how) for the value of the r-th position of the attention contextual vector, where r is the rank of l in $\boldsymbol{V}$.

BERT models use their own word-breaking tokenizers, such that if a word is not in the tokenizer's vocabulary, it is split into parts. Thus, position j in the Rodna-tokenized sentence will point to a token that is different from the token at position j in the BERT-tokenized sentence. To compute attention weights between lemmas in the Rodna-tokenized sentence, we maintain a bidirectional index mapping between the two tokenizations.

Computation of the attention vocabulary $\boldsymbol{V}$ for lemma L across all examples of L involves determining which lemmas l are the most relevant for L in each example of L. Thus, given lemma L at position j in a Rodna-tokenized sentence i, let us introduce some notations first:

- n is the index of a BERT hidden layer (BERT models have between 12 and 24 hidden layers).
- h_m^n is the m-th attention head of hidden layer n (each hidden layer contains 8 to 16 attention heads).
- $h_m^n[j]$ is the *softmaxed* attention column vector of the word at position j in the BERT-tokenized sentence for the head h_m^n.
- $topk(h_m^n[j])$ returns the top k ($k = 3$) list of pairs of lemmas with their attention weights (l, w) that are closest to the lemma L.

We keep track of lemma and attention weight pairs returned by function $topk(h_m^n[j])$ at each hidden level n and attention head m in the BERT model in a dictionary data structure we call D_i. The *keys* of D_i are the lemmas, and the *values* of the keys are lists to which we add the weights that were associated with the lemma keys. $D_i[l]$ is the list of weights for lemma l.

We can select which lemmas of the dictionary D_i are going into the attention vocabulary $\boldsymbol{V}$ in one of the following three ways:

1. For each lemma l in D_i, compute the average of weights $a = \frac{\sum_{w \in D_i[l]} w}{|D_i[l]|}$, create a list F_1 of (l, a) pairs and sort it in descending order by a. We call this method `mean`.
2. For each lemma l in D_i and weight w in $D_i[l]$, create a list F_2 of (l, w) pairs and sort it in descending order by w. Eliminate all pairs (l, w) for which l is in a pair with a bigger weight. We call this method `max`.
3. For each lemma l in D_i, compute the size of its weight list $s = |D_i[l]|$, create a list F_3 of (l, s) pairs and sort it in descending order by s. We call this method `heads`.

Out of any of the lists, F_1, F_2 and F_3, we can select some lemmas to go into the vocabulary $\boldsymbol{V}$. This can be carried out in any of the following three ways:

1. From any of the lists, F_1, F_2, or F_3, we can send the top `p_vocab_topk` (an integer which is a parameter of this algorithm) to vocabulary $\boldsymbol{V}$.
2. From lists F_1 or F_2, we can set a cutoff threshold `p_vocab_cutoff` on the value of the associated weight and send the lemmas appearing with a weight that is bigger or equal to `p_vocab_cutoff`.
3. From any of the lists, F_1, F_2, or F_3, we can construct a global (i.e., for all example sentences) lemma frequency dictionary and select the top `p_vocab_freq` lemmas to constitute the vocabulary $\boldsymbol{V}$.

We note that the WSI algorithm depends on a set of parameters (we used the `p_<name>` notation above) to which we add the parameter method of producing the lists F_1, F_2, or F_3 which we call the `p_vocab_method`. We automatically optimize the values of these parameters by exhaustively searching for the combination of values that maximizes the V-Measure of cluster overlap on the 1000 annotated examples of the adjective "aerian", and the optimal values of these parameters are then used when performing WSI on all other lemmas in our lexical sample (i.e., at runtime, the optimally parametrized WSI algorithm will choose Romanian and English cluster sets that maximize their V-Measure). We detail the automatic search procedure in Section 4.4 using the k-means and agglomerative clustering algorithms from Python's scikit-learn library [27].

4.2. The WSD Algorithm

The idea behind the WSD algorithm is that while clustering is not perfect, if it is reasonably good, we can assume that the majority of example sentences of lemma L from a cluster have the same sense, while the rest are misclassified (have some other sense). If we had a method to sample "positive examples" from the examples that have the expected sense s and "negative examples" from the examples that have some other sense in the cluster, we could train a classifier to tell us if the rest of the cluster examples are positive or negative. Again, if the cluster is reasonably good and the choice of s is correct, we expect that most labels are positive.

Formally, if the WSI algorithm provides a set of disjoint clusters $C = \{c_1, c_2, \ldots, c_k\}$ of senses of lemma L, across all its N example sentences ($\sum_{i=1}^{k} |c_i| = N$), the job of the WSD algorithm is to find the sense identifier s_i that maximizes the conditional probability $P(s_i|c_j) = \frac{P(s_i, c_j)}{P(c_j)}$ for each cluster, where $c_j \in C$. The probability of a cluster c_j is simply the number of example sentences of lemma L that were classified in that cluster, i.e., $P(c_j) = \frac{|c_j|}{N}$, leaving the estimation of joint probability $P(s_i, c_j)$ to the WSD algorithm.

The "sense inventory" S for lemma L is a set of manually sense-tagged example sentences, randomly sampled from the CoRoLa for each lemma L of Table 2 and not overlapping with the example sentences of our lexical sample, in such a way that every valid sense identifier s_i of L is instantiated by at most five example sentences. Strictly speaking, this way of defining our sense inventory will change the designation of our WSD algorithm from "unsupervised" to "semi-supervised", even though the "supervision" is minimal, i.e., we only use at most five sense-tagged example sentences per sense.

For a sense identifier s_i with its example sentences $X = \{x_1^i, x_2^i, x_3^i, x_4^i, x_5^i\}$ and a cluster c_j containing clustered example sentences $Y = \left\{y_1^j, y_2^j, \ldots, y_{|c_j|}^j\right\}$, the WSD algorithm performs the following steps to estimate $P(s_i, c_j)$:

1. For sentences $x_a^i \in X$ and $y_b^j \in Y$, for all $1 \leq a \leq 5$ and $1 \leq b \leq |c_j|$, we extract the BERT embeddings corresponding to the occurrence of lemma L and compute a cosine similarity between them. We sort all pairs $\left(y_b^j, x_a^i\right)$ by the cosine similarity in descending order and keep the top 10% of y_b^j examples with a cosine similarity of

at least 0.7 as "Belonging to sense s_i" and the bottom 10% of y_b^j examples as "Not belonging to sense s_i" as a "train set" of s_i mapping to cluster c_j.

2. We fine-tune the BERT model to classify the remaining 80% of examples in the cluster c_j as either "Belonging to sense s_i" (label 1) or "Not belonging to sense s_i" (label 0). The classifier uses the BERT embedding of lemma L in example sentence y_b^j onto which it stacks a two-neuron, fully connected softmaxed classification layer that is trained (along with the BERT parameters) on the "train set" produced in step 1.
3. From the classification of the remaining 80% of examples in the cluster c_j, we obtain example sentences y_b^j that have been assigned label 0 or label 1. If we count the number of times that label 1 was assigned and divide it by $0.9 \cdot |c_j|$ (10% were already assumed to have label 1 in the train set), we obtain our estimate of $P(s_i, c_j)$.

The BERT model [28,29] with the classifier head was fine-tuned with a learning rate of 10^{-5}, for two epochs, with a batch size of two example sentences. A fresh BERT model is instantiated for every pair of s_i and c_j.

The baseline version of this algorithm is to run on a single cluster containing all examples of lemma L, effectively maximizing $P(s_i, c_1)$.

4.3. A Qualitative Comparison with the Current State-of-the-Art Unsupervised and Knowledge-Based WSD Algorithms

The WSD algorithm introduced in Section 4.2 above is, to the best of our knowledge, the first WSD algorithm that uses the few-shot learning paradigm, a machine learning method that is specific to LLMs. It works for large volumes of data (i.e., at least 500 example sentences per target lemma, needed for fine-tuning BERT models), thus being suitable for performing WSD on large corpora. It only needs example sentences for each known sense of a target lemma, not needing lexical ontologies such as WordNet that are commonplace for knowledge-based WSD algorithms.

The WSI algorithm, on which WSD relies to learn the mapping of sense IDs to cluster IDs, leverages translations to other languages to automatically find how many senses of the target lemma are present in the analyzed sample. This is achieved by synchronously clustering Romanian and English translated example sentences of the target lemma such that the V-Measure of cluster overlapping is maximized. Even though we only use English translations, our previous work [7] suggests that adding translations can only improve this process.

The main advantages and disadvantages of the proposed method compared to the current state-of-the-art unsupervised and knowledge-based WSD presented in Section 2 are as follows:

1. It does not rely on structured sense inventories to function (e.g., WordNet), and it can run on very large corpora. The game theoretic approach proposed in [22] disambiguates all content words in a text simultaneously, thus having to be optimized when run on a corpus with hundreds of millions of words. Furthermore, it uses sense-annotated corpora to initialize the sense distribution vectors of the players, which makes it a semi-supervised WSD approach.
2. It works with any pre-trained and language-specific Transformer-based LLM, as opposed to sense embedding models (e.g., PolyLM [19]) that must be pretrained on very large corpora first.
3. It uses all the attention layers of the BERT model to build a richer contextualized representation of a word, as opposed to algorithms that only use the final BERT hidden state as the contextualized representation [20,21]. In this sense, our attention-based contextual representation is more like the word substitution representation from [18].

4. It may fail to work if the target lemma has a frequency that is less than 100, as there are few examples in the cluster to estimate the sense-to-cluster mapping probability. This is the reason why this algorithm is suited to large and very large corpora.

4.4. Case Study: Adjective "Aerian"

As previously mentioned in Section 3, we manually annotated 1000 examples of the adjective "aerian" with sense IDs to serve as a more reliable gold standard when searching for the combination of parameter values of the WSI algorithm that maximizes the Romanian and English average V-Measure (*VM*). The possible values of the parameters used by the WSI algorithm are as follows:

1. k-means (`kmn`) clustering or agglomerative (`agg`) clustering with the "cosine" distance and the "average" linkage method.
2. Clustering with BERT contextual vectors (`bert`), attention contextual vectors (`attn`) or a concatenation of both (`both`). The following parameters only apply when we do not use `bert`.
3. `p_vocab_method` can be `mean`, `max` or `heads`.
4. `p_vocab_topk` takes values from the list of 1, 2, 5 and 10.
5. `p_vocab_cutoff` takes values from the list of 0.3, 0.5, 0.7 and 0.9.
6. `p_vocab_freq` takes values from the list of 10, 20, 50 and 100.

The automatic search procedure is simply an exhaustive search in the Cartesian product of the sets enumerated at steps 1–6 above, yielding a search space of $2 \times 3 \times 3 \times 4 \times 4 \times 4 = 1152$ parameter value tuples. For each parameter value tuple, the WSI algorithm is run independently on the Romanian examples of the lemma "aerian", and on their English translations, starting from five clusters, the maximum number of senses for lemma "aerian", and going down to two clusters (we know that each lemma in our lexical sample set has at least two senses). If R is the Romanian cluster set, E is the English cluster set and G is the gold-standard cluster set, the parameter value tuple that maximizes $\frac{VM(R,G)+VM(E,G)}{2}$ is the one we will continue to use for the rest of our lemmas in the lexical sample.

Table 5 presents the results of the parameter search procedure for each clustering algorithm and contextual vector type. We used `readerbench/RoBERT-small` [29] for Romanian and `FacebookAI/roberta-base` for English [30]. There are five clusters in the gold-standard cluster set G: $|G| = 5$.

Table 5. Optimum gold-standard Romanian and English average V-Measure with different clustering algorithms and contextual vector types for adjective "aerian".

Clustering	Vector Type	p_vocab_method	p_vocab_topk	p_vocab_cutoff	p_vocab_freq	$\|R\|$	$\|E\|$	VM (%)
kmn	bert	n/a	n/a	n/a	n/a	4	5	7.41
agg	bert	n/a	n/a	n/a	n/a	5	4	3.38
kmn	both	mean	n/a	n/a	20	5	5	9.08
agg	both	mean	n/a	n/a	10	5	4	3.38
kmn	attn	mean	n/a	0.7	n/a	2	3	13.01
agg	attn	mean	n/a	n/a	100	5	4	18.8

The attention contextual vector combined with the agglomerative clustering obtains the best V-Measure against the 1000 annotated sentences of the adjective "aerian". If we note the number of clusters that are created, we see that we obtain five clusters in Romanian, as many as the gold standard has, while we obtain four clusters in English, proving that meaning is conserved by translation. There are two reasons why we do not obtain the same number of clusters in English as in Romanian:

- Translation errors that cause the respective sentences to be reassigned to other clusters or to form new clusters.
- Translation can be a hypernym of our lemma of interest, and thus, it encompasses different senses in the source language.

To prove that a similar number of Romanian and English clusters will favor the V-Measure, Table 6 presents this measure computed for all cluster number pairs from two to five, in Romanian and English, with the best parameter values of the WSI algorithm from Table 5 (i.e., using the agglomerative clustering and attention contextual vectors). We see that the best V-Measure values are for three or four Romanian clusters and four or five English clusters.

Table 6. Romanian-to-English V-Measure with different cluster numbers for adjective "aerian".

	$\|E\| = 2$	$\|E\| = 3$	$\|E\| = 4$	$\|E\| = 5$
$\|R\| = 2$	0.02%	0.01%	0.24%	0.28%
$\|R\| = 3$	0.4%	0.7%	8%	7.47%
$\|R\| = 4$	0.64%	0.95%	7.64%	7.21%
$\|R\| = 5$	0.44%	0.88%	6.28%	6.11%

To put the maximum value of 8% in Table 6 into perspective, here is a breakdown of the gold-standard annotation of the adjective "aerian":

- There were 81 instances of DEX ID '1', which maps to Princeton WordNet's synset "aerial—existing or living or growing or operating in the air".
- There were 869 instances of DEX ID '2', which maps to Princeton WordNet's synset "air—travel via aircraft".
- There were five instances of DEX ID '3', which maps to Princeton WordNet's synset "aerial—characterized by lightness and insubstantiality: as impalpable or intangible as air".
- There were three instances of DEX ID '4', which is the same sense as the Collins Dictionary [31] "absent-minded—forgets things or does not pay attention to what they are doing, often because they are thinking about something else".
- There were 38 instances of DEX ID '5', which maps to Princeton WordNet's synset "respiratory tract, airway—the passages through which air enters and leaves the body".

In Romanian, the three clusters of "aerian" that gave the 8% V-Measure in Table 6 refer to the following senses (by a quick and random inspection of about 10 examples in each cluster):

- There were 4749 instances of DEX ID '2' (this is cluster ID '0'; see Table 7 below for how WSD can correctly map this sense ID to this cluster ID).
- There were 83 instances with a sub-sense of DEX ID '2' related to aerial battles or attacks (cluster ID '1').
- There were 168 instances of DEX ID '1' (cluster ID '2').

Table 7. Romanian sense-to-cluster mapping for adjective "aerian".

	Cluster ID '0'	**Cluster ID '1'**	**Cluster ID '2'**
DEX ID '1'	$P(s = \text{'1'} \mid c = \text{'0'}) = 0.184$	0.283	0.293
DEX ID '2'	**0.301**	0.193	0.22
DEX ID '3'	0.058	0.117	0.006
DEX ID '4'	0.231	0.007	0.087
DEX ID '5'	0.226	**0.4**	**0.393**

In English, the four corresponding clusters from Table 6 refer to the following senses:

- There were 4634 instances of DEX ID '2'.
- There were 148 instances of DEX ID '1'.
- There were 128 instances with a sub-sense of DEX ID '2' related to air forces.
- There were 90 instances with a sub-sense of DEX ID '2' related to aerial rescuing missions.

It is no surprise that the rare sense examples with DEX IDs from '3' to '5' are engulfed by bigger clusters, both in English and in Romanian. But the distribution of the most frequent senses (DEX IDs '2' and '1') is preserved in both Romanian and English, and it is the same as in the gold-standard annotation.

We applied the WSD algorithm on the three Romanian clusters that obtained the 8% V-Measure cluster overlap with English in Table 6. We present, in Table 7, the conditional probabilities $P(s|c)$ computed for each s in DEX IDs '1' to '5' and each c in cluster IDs '0' to '2'. Underlined values mark the correct sense mapping, and the bold values indicate the WSD sense mapping.

We see that the WSD algorithm can correctly map the sense to the cluster, if there are enough examples in the cluster to support the robust estimation of $P(s|c)$. Cluster '0' has 4749 examples, while clusters '1' and '2' have 83 and 168 examples, respectively. If we compute the standard WSD accuracy, because the largest cluster was correctly mapped, we obtain 83.9% on our gold-standard annotation. In terms of F-measure, we obtain a 92.3% F-measure for DEX ID '2' and 0% for all other senses.

With the baseline version of the WSD algorithm, running on a single cluster containing all 5000 examples of the adjective "aerian", we achieve an accuracy of 0.3% because the winning sense ID is not DEX ID '2' but DEX ID '4' (by a small margin; see Table 8), which only has three annotated instances.

Table 8. Romanian baseline sense-to-cluster mapping for adjective "aerian".

	Cluster ID '0'
DEX ID '1'	$P(s = \text{'1'} \mid c = \text{'0'}) = 0.174$
DEX ID '2'	0.23
DEX ID '3'	0.127
DEX ID '4'	**0.244**
DEX ID '5'	0.225

5. Results and Discussion

We ran the optimally parametrized (as shown in Section 4.4) WSI algorithm and both the WSD algorithm and its baseline on all lemmas in our lexical sample (Table 9 presents the results). We include the following information for each lemma:

- Number of Romanian and English clusters determined with the V-Measure overlap method.
- Number of Romanian and English clusters in the 200-example gold standard.
- V-Measure of cluster overlapping.
- Paired F-score overlap [12].
- WSD accuracy and baseline WSD accuracy.

Table 9. Romanian WSI and WSD results on the lexical sample (shaded values are worse).

Lemma	$\|R\|$	$\|E\|$	VM (%)	FS (%)	Gold $\|R\|$	WSD acc. (%)	BL WSD acc. (%)
Nouns							
pondere	3	3	0.23	97.2	3	86	86
caiet	2	2	0	97.3	2	80.5	80.5
incintă	3	3	1.43	85.3	3	63.4	43.8
relief	3	3	0.4	90.9	3	8.8	38.9
codru	2	2	0.07	97.7	2	97.2	97.2
puț	2	2	0.35	96.4	2	57.5	58
papuc	2	2	1	91.9	2	99	99
brumă	3	3	40.74	84.5	3	29.4	46
ansă	4	4	2.56	75.8	4	65.3	93.5
săpuneală	3	2	11.5	72	3	92	10
Verbs							
abilita	2	2	0.28	92.4	2	99	99
disputa	3	3	0.23	95.5	3	61.6	29.3
dispera	2	2	0.15	97.8	2	89.3	89.3
recepționa	2	2	0.1	97.5	2	33	33
răci	3	3	0.13	94.1	3	71.4	71.4
depărta	3	3	0.46	91.8	3	7.5	69.8
gripa	2	2	0.87	92.6	2	88.5	11.5
înseta	2	2	0.1	98.5	2	22.2	77.8
parveni	2	2	32.75	93.4	4	27.6	20.9
înfrăți	2	2	100	100	2	99	99
Adjectives							
oficial	3	3	0	99.9	3	93.5	93.5
unic	2	2	0	99.9	2	94.5	94.5
cult	3	3	0.49	63.8	3	59.6	59.6
aerian	3	4	8	89.7	5	83.9	0.3
conform	2	2	0	65.7	2	0.5	42.9
reprezentativ	2	2	0	99.3	2	48.5	53.5
verbal	2	2	1.72	72.5	2	76.3	76.3
vegetal	3	3	2.72	93.7	3	88.5	88.5
sectorial	2	2	0	99.5	2	99	99
liric	4	4	0.36	92.4	4	2.5	0.5
Adverbs							
aci	2	2	0.32	96.4	2	70.3	73.7
orbește	2	2	0.47	95.4	2	88.9	91.1
zdravăn	2	2	0.06	96.4	2	52	48
omenește	2	2	0.06	95.1	2	38.1	38.1
Average	n/a	n/a	n/a	n/a	n/a	**63.9**	62.1

With respect to WSI results, we see that we have many cases with a V-Measure that is below 1%, in some cases being even 0%. At the same time, the paired F-score (column FS in Table 9) exceeds 90%. This is explained by the fact that in all respective cases, 99% percent of the examples are placed in a cluster, with the remaining 1% being distributed in the other clusters. This is the right decision for the WSI algorithm, given the fact that the gold-standard annotation, in all these cases, has roughly the same example distribution, albeit not that skewed: 90% in one cluster vs. 10% in the other clusters.

As noted by Manandhar et al. [12], the V-Measure is 0% for a single cluster, hence the result we obtained when 99% of examples are placed in a single cluster. They also observe that the V-Measure favors systems that produce more clusters, but the measure does not increase monotonically with the number of clusters. On the other hand, the paired F-score (which essentially measures how many example pairs are placed in the same cluster in the predicted clusters and in the gold-standard clusters) favors systems that produce a

lower number of clusters. We can also observe this behavior in Table 9 above, where when we have two clusters, the V-Measure is usually lower than 1% and the paired F-score is greater than 90%. Ideally, one would want a maximum V-Measure and paired F-score to achieve perfect clustering, and we only have one such instance, for the lemma "înfrăți", with clusters in Romanian and English being identical while having the same distribution as the gold-standard clusters: 99% of examples in one cluster and 1% in the other clusters.

With respect to WSD, we see that, on average, the WSI algorithm helps the WSD algorithm be 1.8% better than the WSD algorithm running on a single cluster (the baseline). The WSD algorithm could not improve the baseline disambiguation for 16 lemmas, and, in all cases, the cluster distribution was heavily skewed towards the most frequent sense, which the baseline WSD algorithm is able to find quite reliably (we did not have this information in our sense inventory prior to running WSD).

To evaluate the impact of translation accuracy on WSI and WSD, we manually inspected a random sample of 100 translated examples for each lemma in our lexical sample, judging the translation equivalent of the targeted lemma in each example (Table 10 contains the translation accuracy evaluations). There are ways in which one can automatically evaluate machine translation in an unsupervised manner (i.e., without having access to reference translations) [32,33] but, because we use the attention weights of tokens relating to the lemma of interest and its translation, we primarily need to know how this lemma was translated. Otherwise, just by eyeballing the translations produced by Mistral 7B, we can say that the translations are of good quality, generally speaking.

Table 10. English translation accuracy (the English translation in the parentheses is of the most frequent correct translation).

Nouns		Verbs		Adjectives		Adverbs	
pondere (weight)	92.8%	abilita (ability)	96.3%	oficial (official)	73.5%	aci (here)	46.3%
caiet (notebook)	90.8%	disputa (dispute)	93.4%	unic (unique)	63.1%	orbește (blindly)	11.1%
incintă (premise)	83%	dispera (desperate)	85.8%	cult (culture)	19.4%	zdravăn (healthy)	43.1%
relief (terrain)	9.3%	recepționa (receive)	95.2%	aerian (air)	92.8%	omenește (human)	72.1%
codru (forest)	44.8%	răci (cool)	83.9%	conform (consistent)	31.7%		
puț (well)	83.6%	depărta (away)	84.7%	reprezentativ (representative)	94.8%		
papuc (slipper)	59.1%	gripa (sick)	69%	verbal (verbal)	90.7%		
brumă (frost)	40.5%	înseta (craving)	79.5%	vegetal (vegetable)	93%		
ansă (loop)	42.5%	parveni (reach)	78.9%	sectorial (sectorial)	96%		
săpunea (soap)	26%	înfrăți (friend)	80.5%	liric (lyrical)	88.4%		

Averaging the translation accuracies for nouns, verbs, adjectives and adverbs from Table 10 and comparing them with the average V-Measures of nouns, verbs, adjectives and adverbs from Table 9, we can see that the translation accuracy correlates positively with V-Measure, except for adjectives (see Table 11). This is experimental evidence that translation quality directly influences Romanian and English clustering, a fact that is most obvious in the case of adverbs with an average V-Measure of 0.2 and an average translation accuracy of 43.1%.

Table 11. Average measures for nouns, verbs, adjectives and adverbs of the lexical sample.

Averages	Nouns	Verbs	Adjectives	Adverbs
Translation accuracy	57.2%	84.7%	74%	43.1%
V-Measure	5.8	13.5	1.3	0.2
WSD accuracy	67.9%	59.9%	64.6%	62.3%
Baseline WSD accuracy	65.2%	60.1%	60.8%	62.7%
Number of senses in GS	2.7	2.5	2.8	2

Adjectives, on the other hand, seem to defy the correlation of high translation quality with high clustering quality. This is explainable by the fact that their English translations lexicalize different Romanian meanings with the same English translation equivalent, thus not differentiating the English embeddings enough (the average V-Measure of adjectives is only 1.3, but the average number of senses in the gold standard is the highest: 2.8).

In theory, bigger Romanian models (i.e., with more parameters) should perform better on the WSD task. To put this statement to the test, we selected four lemmas (one in each grammatical category) for which the baseline WSD accuracy was much higher with the `readerbench/RoBERT-small` model—noun "relief", verb "depărta", adjective "conform" and adverb "aci"—and we ran both the WSD and the baseline WSD algorithms with the bigger versions of `readerbench/RoBERT-small` model—`readerbench/RoBERT-base` [34] and `readerbench/RoBERT-large` [35]. Furthermore, to verify if a multilingual BERT model is as effective as the Romanian-specific models, we also ran the WSD and the baseline WSD algorithms for the four lemmas mentioned above, with the `FacebookAI/xlm-roberta-large` model [36]. Table 12 presents the results.

Table 12. WSD and baseline (BL) WSD accuracy results with different BERT models (shaded values are worse for a model, bolded values are the best on the row and underlined values are close to the best ones).

Lemma/POS	RoBERT-Small		RoBERT-Base		RoBERT-Large		XLM-RoBERTa-Large	
	WSD	BL WSD	WSD	BL WSD	WSD	BL WSD	WSD	BL WSD
relief/n	8.8%	**38.9%**	**38.9%**	4.7%	38.3%	4.7%	0%	0%
depărta/v	7.5%	**69.8%**	66.8%	24.6%	68.8%	24.6%	68.8%	0%
conform/a	0.5%	42.9%	56.1%	42.9%	42.4%	**56.6%**	42.9%	0%
aci/r	70.3%	**73.7%**	70.3%	**73.7%**	70.3%	25.7%	0%	0%

Table 12 shows that bigger language models can bring benefits, but there is no monotonic increase in WSD accuracy when we go up the ladder of language model complexity. We can tie the results obtained by RoBERT-small for "relief" and "aci", we can improve on the accuracy of "conform", but we can only come close, with respect to accuracy, to "relief". Another observation is that with bigger language models, we can consistently beat the baseline algorithm. WSD with XLM-RoBERTa-large is not able to function properly in most cases (0% accuracy) because it cannot find positive examples when computing the cosine function between sense example sentences and cluster example sentences, thus estimating $P(s_i, c_j)$ to be 0. This is conclusive proof that multilingual language models are poor substitutes for language-specific ones.

6. Conclusions

The paper shows that (extremely) weakly supervised WSD is possible and feasible with the advent of LLMs that store a wealth of information into their attention layers. Our average accuracy of 64% on 6716 examples of all types of content words is very close to the accuracies of supervised approaches from the SensEval era [11], and this is made possible by the ability of the Transformer models to learn a lot by just seeing a few examples (the

few-shot learning paradigm). Future work includes trying more BERT flavors or even generative models such as Llama or OLMo, but, from our limited experimentation to date, there is no monotonic increase in WSD performance just by swapping in a larger language model. The most obvious future course of action is to develop parallel and synchronous Romanian and English WSD, executed on Romanian and English clusters that optimize the V-Measure.

Ideas put forth in [7] use translation equivalents to build cross-language contextual vectors, but nowadays, we can use the attention layers of BERT models to upgrade that approach. To further restrict the semantic space of the target Romanian lemma, as suggested in [7], we can make use of automatic translations in languages from groups other than Romance, such as Hellenic or Slavic, provided that the automatic translation quality from Romanian into these languages (e.g., Greek or Bulgarian) is good enough. As explained in [7], this approach has the potential to "resolve" the sense of a Romanian target lemma, because, with multiple translations, each sense of the Romanian target lemma is represented by the same (or very similar) vector of translation equivalents. This intuition suggests that translation should be made into languages from different language groups, thus making English a good first choice, in addition to the argument that the translations from and to English are the best due to the widespread availability of training data.

Author Contributions: Conceptualization, R.I.; methodology, R.I.; software, R.I., V.P. and V.B.; validation, V.B.M., E.I. and M.M.; formal analysis, R.I.; investigation, R.I., V.P. and D.T.; resources, V.B.M., E.I. and M.M.; data curation, V.B.M., E.I. and M.M.; writing—original draft preparation, R.I.; writing—review and editing, V.P., V.B.M., E.I., M.M., V.B. and D.T.; supervision, R.I. and D.T.; project administration, R.I. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Due to copyright issues, the lexical sample dataset cannot be made public, but it is available from the authors upon completing the required release form. The Python 3 code is available by request at radu@racai.ro.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Vaswani, A.; Jones, L.; Shazeer, N.; Parmar, N.; Gomez, A.N.; Uszkoreit, J.; Kaiser, Ł.; Polosukhin, I. Attention Is All You Need. In Proceedings of the 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA, 4–9 December 2017.
2. Hello GPT-4o. Available online: https://openai.com/index/hello-gpt-4o/ (accessed on 10 October 2024).
3. Gemini Models. Available online: https://deepmind.google/technologies/gemini/ (accessed on 10 October 2024).
4. Schütze, H. Automatic word sense discrimination. *Comput. Linguist.* **1998**, *24*, 97–123.
5. Song, X.; Salcianu, A.; Song, Y.; Dopson, D.; Zhou, D. Fast WordPiece Tokenization. *arXiv* **2020**, arXiv:2012.15524.
6. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Proceedings of the NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019.
7. Tufiș, D.; Ion, R.; Ide, N. Fine-Grained Word Sense Disambiguation Based on Parallel Corpora, Word Alignment, Word Clustering and Aligned Wordnets. In Proceedings of the 20th International Conference on Computational Linguistics COLING 2004, Geneva, Switzerland, 23–27 August 2004.
8. Tufiș, D.; Barbu Mititelu, V. The Lexical Ontology for Romanian. In *Language Production, Cognition, and the Lexicon, Series Text, Speech and Language Technology*; Gala, N., Rapp, R., Bel-Enguix, N., Eds.; Springer: Berlin/Heidelberg, Germany, 2014; Volume 48.
9. Fellbaum, C. (Ed.) *WordNet: An Electronic Lexical Database*; MIT Press: Cambridge, MA, USA, 1998.
10. Wang, Y.; Yao, Q.; Kwok, J.T.; Ni, L.M. Generalizing from a Few Examples: A Survey on Few-shot Learning. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–34. [CrossRef]
11. Raganato, A.; Camacho-Collados, J.; Navigli, R. Word Sense Disambiguation: A Unified Evaluation Framework and Empirical Comparison. In Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers, Valencia, Spain, 3–7 April 2017.

12. Manandhar, S.; Klapaftis, I.P.; Dligach, D.; Pradhan, S.S. SemEval-2010 Task 14: Word Sense Induction & Disambiguation. In Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010, Uppsala, Sweden, 15–16 July 2010.
13. Word Sense Induction. Available online: https://paperswithcode.com/task/word-sense-induction (accessed on 17 October 2024).
14. Bartunov, S.; Kondrashkin, D.; Osokin, A.; Vetrov, D. Breaking Sticks and Ambiguities with Adaptive Skip-gram. *PMLR* **2016**, *51*, 130–138.
15. Sun, Y.; Rao, N.; Ding, W. A Simple Approach to Learn Polysemous Word Embeddings. *arXiv* **2017**, arXiv:1707.01793.
16. Huang, E.H.; Socher, R.; Manning, C.D.; Ng, A.Y. Improving word representations via global context and multiple word prototypes. In Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics Volume 1: Long Papers, Jeju, Republic of Korea, 8–14 July 2012.
17. Amplayo, R.K.; Hwang, S.; Song, M. AutoSense Model for Word Sense Induction. In Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19), Honolulu, HI, USA, 27 January–1 February 2019.
18. Eyal, M.; Sadde, S.; Taub-Tabib, H.; Goldberg, Y. Large Scale Substitution-based Word Sense Induction. In Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics Volume 1: Long Papers, Dublin, Ireland, 22–27 May 2022.
19. Ansell, A.; Bravo-Marquez, F.; Pfahringer, B. PolyLM: Learning about Polysemy through Language Modeling. In Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, Online, 19–23 April 2021.
20. Chawla, A.; Mulay, N.; Bishnoi, V.; Dhama, G.; Singh, A.K. A Comparative Study of Transformers on Word Sense Disambiguation. In *Neural Information Processing. ICONIP 2021. Communications in Computer and Information Science*; Mantoro, T., Lee, M., Ayu, M.A., Wong, K.W., Hidayanto, A.N., Eds.; Springer: Cham, Switzerland, 2021; Volume 1516.
21. Vandenbussche, P.-Y.; Scerri, T.; Daniel, R., Jr. Word Sense Disambiguation with Transformer Models. In Proceedings of the 6th Workshop on Semantic Deep Learning (SemDeep-6), Online, 8 January 2021.
22. Tripodi, R.; Navigli, R. Game Theory Meets Embeddings: A Unified Framework for Word Sense Disambiguation. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, Hong Kong, China, 3–7 November 2019.
23. Academia Română, Institutul de Lingvistică "Iorgu Iordan—Al. Rosetti". *DEX—Dicționarul Explicativ al Limbii Române*; Univers Enciclopedic: București, România, 2016.
24. Tufiș, D.; Barbu Mititelu, V.; Irimia, E.; Păiș, V.; Ion, R.; Diewald, N.; Mitrofan, M.; Onofrei, M. Little Strokes Fell Great Oaks. Creating CoRoLA, The Reference Corpus of Contemporary Romanian. *RRL* **2019**, *64*, 227–240.
25. Ion, R. (Ed.) *Evaluating and User-Testing Rodna, A New Romanian Text Processing Pipeline*; Research report; Romanian Academy: Bucharest, Romania, 2022.
26. Mistral. Available online: https://ollama.com/library/mistral (accessed on 24 October 2024).
27. Scikit-Learn. Available online: https://scikit-learn.org/1.5/index.html (accessed on 22 October 2024).
28. Masala, M.; Ruseti, S.; Dascalu, M. RoBERT—A Romanian BERT Model. In Proceedings of the 28th International Conference on Computational Linguistics, Barcelona, Spain, 8–13 December 2020.
29. Readerbench/RoBERT-Small. Available online: https://huggingface.co/readerbench/RoBERT-small (accessed on 22 October 2024).
30. FacebookAI/Roberta-Base. Available online: https://huggingface.co/FacebookAI/roberta-base (accessed on 22 October 2024).
31. Collins Dictionary. Available online: https://www.collinsdictionary.com/dictionary/english (accessed on 23 October 2024).
32. Elmakias, I.; Vilenchik, D. An Oblivious Approach to Machine Translation Quality Estimation. *Mathematics* **2021**, *9*, 2090. [CrossRef]
33. Moosa, I.M.; Zhang, R.; Yin, W. MT-Ranker: Reference-free machine translation evaluation by inter-system ranking. In Proceedings of the 12th International Conference on Learning Representations, ICLR 2024, Vienna, Austria, 7–11 May 2024.
34. Readerbench/RoBERT-Base. Available online: https://huggingface.co/readerbench/RoBERT-base (accessed on 6 January 2025).
35. Readerbench/RoBERT-Large. Available online: https://huggingface.co/readerbench/RoBERT-large (accessed on 6 January 2025).
36. FacebookAI/xlm-Roberta-Large. Available online: https://huggingface.co/FacebookAI/xlm-roberta-large (accessed on 6 January 2025).