Game Theory Meets Embeddings: a Unified Framework for Word Sense Disambiguation

Rocco Tripodi
Ca' Foscari University of Venice
rocco.tripodi@unive.it

Roberto Navigli
Sapienza University of Rome
navigli@di.uniroma1.it

Abstract

Game-theoretic models, thanks to their intrinsic ability to exploit contextual information, have shown to be particularly suited for the Word Sense Disambiguation task. They represent ambiguous words as the players of a non-cooperative game and their senses as the strategies that the players can select in order to play the games. The interaction among the players is modeled with a weighted graph and the payoff as an embedding similarity function, which the players try to maximize. The impact of the word and sense embedding representations in the framework was tested and analyzed extensively: experiments on standard benchmarks show state-of-art performances and different tests hint at the usefulness of using disambiguation to obtain contextualized word representations.

1 Introduction

Word Sense Disambiguation (WSD), the task of linking the appropriate meaning from a sense inventory to words in a text, is an open problem in Natural Language Processing (NLP). It is particularly challenging because it deals with the semantics of words and, by their very nature, words are ambiguous and can be used with different meanings in different situations. Among the key tasks aimed at enabling Natural Language Understanding (Navigli, 2018), WSD provides a basic, solid contribution since it is able to identify the intended meaning of the words in a sentence (Kim et al., 2010).

WSD can be seen as a classification task in which words are the objects to be classified and senses are the classes into which the objects have to be classified (Navigli, 2009); therefore it is possible to use supervised learning techniques to solve the WSD problem. One drawback with this idea is that it requires large amounts of data that are difficult to obtain. Furthermore, in the WSD context, the production of annotated data is even more complicated and excessively time-consuming compared to other tasks. This arises because of the variability in lexical use. Furthermore, the number of different meanings to be considered in a WSD task is in the order of thousands, whereas classical classification tasks in machine learning have considerably fewer classes.

We decided to adopt a semi-supervised approach to overcome the knowledge acquisition bottleneck and innovate the strand of research introduced by Tripodi and Pelillo (2017). These researchers developed a flexible game-theoretic WSD model that exploits word and sense similarity information. This combination of features allows the textual coherence to be maintained: in fact, in this model the disambiguation process is relational, and the sense assigned to a word must always be compatible with the senses of the words in the same text. It can be seen as a constraint satisfaction model which aims to find the best configuration of senses for the words in context. This is possible because the payoff function of the games is modeled in a way in which, when a game is played between two players, they are emboldened to select the senses that have the highest compatibility with the senses that the co-player is choosing. Another appealing feature of this model is that it offers the possibility to configure many components of the system: it is possible to use any word and sense representation; also, one can model the interactions of the players in different ways by exploiting word similarity information, the syntactic structure of the sentence and the importance provided by specific relations. Furthermore, it is possible to use different priors on the sense distributions and to use different game dynamics to find the equilibrium state of the model. Traditional WSD methods have only some of these

Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, pages 88–99,
Hong Kong, China, November 3–7, 2019.

properties.

The main difference between our model and the model proposed by Tripodi and Pelillo (2017) is that they did not use state-of-the-art models for word and sense representations. They used word co-occurrence measures for word similarity and *tf-idf* vectors for sense similarity, resulting in sparse graphs in which nodes can be disjoint or some semantic area is not covered. Instead, we are advocating the use of dense vectors, which provide a completely different perspective not only in terms of representation but also in terms of dynamics. Each player is involved in many more games and this affects the computation of the payoffs and the convergence of the system. The interaction among the players is defined in a different way and the priors are modeled with a more realistic distribution to avoid the skewness typical of word sense distributions. Furthermore, our model is evaluated on recent standard benchmarks, facilitating comparison with other models.

The main contributions of this paper are as follows:

1. the release of a general framework for WSD;
2. the evaluation of different word and sense embeddings;
3. state-of-the-art performances on standard benchmarks (in different cases performing better than recent supervised models);
4. the use of disambiguated sense vectors to obtain contextualized word representations.

2 Word Sense Disambiguation

WSD approaches can be divided into two main categories: supervised, which require human intervention in the creation of sense-annotated datasets, and the so-called knowledge-based approaches (Navigli, 2009), which require the construction of a task-independent lexical-semantic knowledge resource, but which, once that work is available, use models that are completely autonomous.

As regards supervised systems, a popular system is *It makes sense* (Zhong and Ng, 2010), a model which takes advantage of standard WSD features such as POS-tags, word co-occurrences, and collocations and creates individual support vector machine classifiers for each ambiguous word. Newer supervised models exploit deep neural networks and especially long short-term memory (LSTM) networks, a type of recurrent neural network particularly suitable for handling arbitrary-length sequences. Yuan et al. (2016) proposed a deep neural model trained with large amounts of data obtained in a semi-supervised fashion. This model was re-implemented by Le et al. (2018), reaching comparable results with a smaller training corpus. Raganato et al. (2017) introduced two approaches for neural WSD using models developed for machine translation and substituting translated words with sense-annotated ones. A recent work that combines labeled data and knowledge-based information has been proposed by Luo et al. (2018). Uslu et al. (2018) proposed *fastSense*, a model inspired by *fastText* (Joulin et al., 2017) which – rather than predicting context words – predicts word senses.

Knowledge-based models, instead, exploit the structural properties of a lexical-semantic knowledge base, and typically use the relational information between concepts in the semantic graph together with the lexical information contained therein (Navigli and Lapata, 2010). A popular algorithm used to select the sense of each word in this graph is PageRank (Page et al., 1999) that performs random walks over the network to identify the most important nodes (Haveliwala, 2002; Mihalcea et al., 2004; De Cao et al., 2010). An extension of these models was proposed by Agirre et al. (2014) in which the Personalized PageRank algorithm is applied. Another knowledge-based approach is Babelfy (Moro et al., 2014), which defines a semantic signature for a given context and compares it with all the candidate senses in order to perform the disambiguation task. Chaplot and Salakhutdinov (2018) proposed a method that uses the whole document as the context for the words to be disambiguated, exploiting topical information (Ferret and Grau, 2002). It models word senses using a variant of the Latent Dirichlet Allocation framework (Blei et al., 2003), in which the topic distributions of the words are replaced with sense distributions modeled with a logistic normal distribution according to the frequencies obtained from WordNet.

3 Word and Sense Embeddings

A good machine-interpretable representation of lexical features is fundamental for every NLP system. A system's performance, however, depends

on the quality of the input representations. Furthermore, the inclusion of semantic features, in addition to lexical ones, has been proven effective in many NLP approaches (Li and Jurafsky, 2015).

Word embeddings, the current paradigm for lexical representation of words, were popularized with word2vec (Mikolov et al., 2013). The main idea is to exploit a neural language model which learns to predict a word occurrence given its surroundings. Another well-known word embedding model was presented by Pennington et al. (2014), which shares the idea of word2vec, but with the difference that it uses explicit latent representations obtained from statistical calculation on word co-occurrences. However, all word embedding models share a common issue: they cannot capture polysemy since they conflate the various word senses into a single vector representation. Several efforts have been presented so far to deal with this problem. SensEmbed (Iacobacci et al., 2015) uses a knowledge-based disambiguation system to build a sense-annotated corpus that, in its turn, is used to train a vector space model for word senses with word2vec. AutoExtend (Rothe and Schütze, 2015), instead, is initialized with a set of pre-trained word embeddings, and induces sense and synset vectors in the same semantic space using an autoencoder. The vectors are induced by constraining their representation given the assumption that synsets are sums of their lexemes. Camacho-Collados et al. (2015) presented NASARI, an approach that learns sense vectors by exploiting the hyperlink structure of the English Wikipedia, linking their representations to the semantic network of BabelNet (Navigli and Ponzetto, 2012). More recent works, such as ELMo (Peters et al., 2018) and BERT (Devlin et al., 2019), are based on language models learned using complex neural network architectures. The advantage of these models is that they can produce different representations of words according to the context in which they appear.

4 Game Theory and Game Dynamics

In this work we take a different approach to WSD by employing a model based on game theory (GT). This discipline was introduced by Neuman and Morgenstern (1944) in order to develop a mathematical framework able to model the essentials of decision making in interactive situations. In its *normal-form* representation (Weibull, 1997), it consists of a finite set of players $N = (1, .., n)$, a finite set of pure strategies $S_i = \{1, ..., m_i\}$ for each player $i \in N$, and a payoff (utility) function $u_i : S \rightarrow \mathbb{R}$, that associates a payoff with each combination of strategies in $S = S_1 \times S_2 \times ... \times S_n$. A fundamental assumption in game theory is that each player i tries to maximize the value of u_i. Furthermore, in *non-cooperative games* the players choose their strategies independently, considering what choices other players can make and trying to find the best response to the strategy of the co-players.

A player i, in addition to playing single (pure) strategies from S_i, can also use *mixed strategies*, that are probability distributions over pure strategies. A mixed strategy over S_i is defined as a vector $\mathbf{x_i} = (x_1, \ldots, x_{m_i})$, such that $x_j \geq 0$ and $\sum x_j = 1$. Each mixed strategy corresponds to a point in the simplex Δ_m, whose corners correspond to pure strategies. The intuition is that player i randomises over strategies according to the probabilities in $\mathbf{x_i}$. Each mixed strategy profile lives in the mixed strategy space of the game, given by the Cartesian product $\Theta = \Delta_{m_1} \times \Delta_{m_2} \times \cdots \times \Delta_{m_n}$.

In a *two-player game*, a strategy profile can be defined as a pair $(\mathbf{x}_i, \mathbf{x}_j)$. The expected payoff for this strategy profile is computed as:

$$u(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \cdot A_{ij}\mathbf{x}_j$$

where A_{ij} is the $m_i \times m_j$ payoff matrix between players i and j.

In evolutionary game theory (Weibull, 1997), the games are played repeatedly and the players update their mixed strategy distributions over time until no player can improve the payoff obtained with the current mixed strategy. This situation corresponds to the equilibrium of the system.

The payoff corresponding to the h-th pure strategy is computed as:

$$u(x_i^h) = x_i^h \cdot \sum_{j=1}^{n_i} (A_{ij}\mathbf{x}_j)^h \quad (1)$$

It is important to note here that the payoff in Equation 1 is additively separable, in fact, the summation is over all the n_i players with whom i is playing the games. The average payoff of player i is calculated as:

$$u(\mathbf{x}_i) = \sum_{h=1}^{m_i} u(x_i^h) \quad (2)$$

To find the Nash equilibrium of the game it is common to use the discrete time version of the *replicator dynamics* equation (Weibull, 1997) for each player $i \in N$,

$$x_i^h(t+1) = x_i^h(t)\frac{u(x_i^h)}{u(\mathbf{x}_i)} \qquad \forall\, h \in \mathbf{x}_i \qquad (3)$$

This equation allows better than average strategies to grow at each iteration. It can be considered as an *inductive learning* process, in which the players learn from past experiences how to play their best strategy. We note that each player optimizes their individual strategy space, but this operation is done according to what other players simultaneously are doing, so the local optimization is the result of a global process.

Game-theoretic models are appealing because they are versatile, interpretable and have a solid mathematical foundation. Furthermore, it is always possible to find the Nash equilibrium in non-cooperative games in mixed strategies (Nash, 1951). In fact, starting from an interior point of Θ, a point $\mathbf{x}$ is a Nash equilibrium only if it is the limit of a trajectory of Equation 3 (Weibull, 1997).

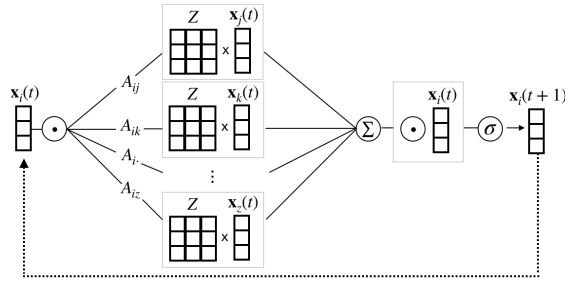

Figure 1: Generic scheme of the model. $\cdot$, $\times$ and σ refer to elementwise multiplication, matrix multiplication and normalization, respectively.

5 The Model

The model used in this paper, Word Sense Disambiguation Games (*WSDG*), was introduced by Tripodi and Pelillo (2017). It is based on graph-theoretic principles to model the geometry of the data and on game theory to model the learning algorithm which disambiguates the words in a text. It represents the words as the players of a non-cooperative game and their senses as the strategy that the players can select in order to play the games. The players are arranged in a graph whose edges determine the interactions and carry word similarity information. The payoff matrix is encoded as a sense similarity function. The players play the games repeatedly and – at each iteration – update their strategy preferences according to what strategy has been effective in previous games. These preferences, as introduced previously, are encoded as a probability distribution over strategies (senses).

Formally, for a text T we select its content words $W = (1, \ldots, n)$ as the players of the game $I = (1, \ldots, n)$. For each word we use a knowledge base to determine its possible senses. Each sense is represented as a strategy that the player can select from the set $S_i = \{1, ..., m_i\}$, where m_i is the number of senses of word w_i. The set of all different senses in the text, $C = \{1, ..., m\}$, is the strategy space of the games. The strategy space is modeled, for each player, as a probability distribution, $\mathbf{x}_i$, of length m. It takes non-zero values only on the entries corresponding to the elements of S_i. It can be initialized with a normal distribution in the case of unsupervised learning or with information obtained from sense-labeled corpora in the case of semi-supervised learning.

The payoff of a game depends on a payoff matrix Z in which the rows are indexed according to the strategies of player i and the columns according to the strategies of player j. Its entries $Z_{r,t}$ are the payoff obtained when player i selects strategy r and player j selects strategy t. It is important to note here that the payoff of a game does not depend on the single strategy taken individually by a player, but always by the combination of two simultaneous actions. In WSD this means that the sense selected by a word influences the choices of the other words in the text and this allows the textual coherence to be maintained.

The disambiguation games to build a payoff function require: a word similarity matrix A, a sense similarity matrix Z and a sense distribution $\mathbf{x}_i$ for each player i. A models the players' interactions, so that similar players play together and the more similar they are the more reciprocal influence they have. It can be interpreted as an attention mechanism (Vaswani et al., 2017) since it weights the payoffs. Z is used to create the payoff matrices of the games so that the more similar the senses of the words are the more the corresponding players are encouraged to select them, since they give a high payoff. A and Z are obtained by computing vector representations of word and sense (see Section 3) and then calculating their pairwise sim-

ilarity.

The strategy space of each player, i, is represented as a column vector of length m. It is initialized with:

$$x_i^h = \begin{cases} |m_i|^{-1} & \text{if sense } h \text{ is in } S_i, \\ 0 & \text{otherwise.} \end{cases} \quad (4)$$

This initialization is used in the case of unsupervised WSD, since it does not use information from sense-tagged corpora. If instead this information is available, $|m_i|^{-1}$ in Equation 4 is substituted with the frequency of the corresponding sense and then $\mathbf{x}_i$ is normalized in order to sum up to one.

Once these sources of information are computed, the *WSDG* are run by using the replicator dynamic equation (Taylor and Jonker, 1978) in Equation 3, where the payoff of strategy h for player i is calculated as:

$$u(x_i^h) = x_i^h \cdot \sum_{j=1}^{n_i} (A_{ij} Z \mathbf{x}_j)^h \quad (5)$$

where n_i are the neighbours of player i as in the graph A. The average payoff is calculated as:

$$u(\mathbf{x}_i) = \sum_{h=1}^{m_i} u(x_i^h) \quad (6)$$

The complexity of *WSDG* scales linearly with the number of words to be disambiguated. Differently from other models based on PageRank, it is possible to disambiguate all the words at the same time. As an example, *WSDG* can disambiguate 200 words (1650 senses) in 7 seconds, on a single CPU core. A generic representation of the model is proposed in Figure 1.

Implementation details The cosine similarity was used as similarity measure for both words and senses. The A matrix was treated as the adjacency matrix of an undirected weighted graph and, to reduce the complexity of the model, the edges with weight lower than 0.1 were removed. The symmetric normalized Laplacian of this graph was calculated as $D^{-1/2}AD^{-1/2}$, where D is the degree matrix of graph A. Since the algorithm operates on an entire text, local information is added to matrix A. The mean value of the matrix is added to the $\lceil log(n) \rceil$ cells on the left of the main diagonal. For BERT, this operation was replaced with its attention layer, adding to matrix A the mean attention distribution of all the heads of the last layer. The choice of the last layer is motivated by the fact that it stores semantic information and its attention distributions have high entropy (Clark et al., 2019). The first singular vector was removed from A in the case of word vectors whose length exceeded 500. This was done to reduce the redundancy of the representations in line with Arora et al. (2017). The distributions for each $\mathbf{x}$ were computed according to SemCor (Miller et al., 1993) and normalized using the softmax function. The replicator dynamics were run until a maximum number of iterations was reached (100) or the difference between two consecutive iterations was below a small threshold (10^{-3}), calculated as $\sum_{i=1}^{n} \|\mathbf{x_i}(t-1) - \mathbf{x_i}(t)\|$. The code of the model is available at https://github.com/roccotrip/wsd_games_emb.

6 Evaluation

The evaluation of our model was conducted using the framework proposed by Raganato et al. (2017). This consists of five datasets which were unified to the same WordNet 3.0 inventory: Senseval-2 (S2), Senseval-3 (S3), SemEval-2007 (SE7), SemEval-2013 (SE13) and SemEval-2015 (SE15). These datasets have in total 26 texts and 10,619 words to be disambiguated. Our objective was to test our game-theoretic model with different settings and to evaluate its performances. To this end we performed experiments comparing 16 different sets of pretrained word embeddings and 7 sets of sense embeddings.

Word embeddings As word embedding models we included 4 pre-word2vec models: the *hierarchical log-bilinear* model (Mnih and Hinton, 2007, HLBL), a probabilistic linear neural model which aims to predict the embedding of a word given the concatenation of the previous words; CW (Collobert and Weston, 2008), an embeddings model with a deep unified architecture for multitask NLP; Distributional Memory (Baroni and Lenci, 2010, DM), a semantically enriched count-based model; leskBasile (Basile et al., 2014), a model based on Latent Semantic Analysis reduced via Singular-Value Decomposition; 3 models obtained with word2vec: GoogleNews, a set of 300-dimensions vectors trained with the Google News dataset; BNC-*, vectors of different dimensions trained on the British National Corpus including POS information during training; and w2vR, trained with word2vec on the 2014

dump of the English Wikipedia, enriched with retrofitting (Faruqui et al., 2015), a technique to enhance pre-trained embeddings with semantic information. The enrichment was performed using retrofitting's best configuration, based on the Paraphrase Database (Ganitkevitch et al., 2013, PPDB). We also tested GloVe (Pennington et al., 2014), trained with the concatenation of the 2014 dump of the English Wikipedia and Gigaword 5, and fastText (Bojanowski et al., 2017) trained on Wikipedia 2017, UMBC corpus and the statmt.org news dataset.

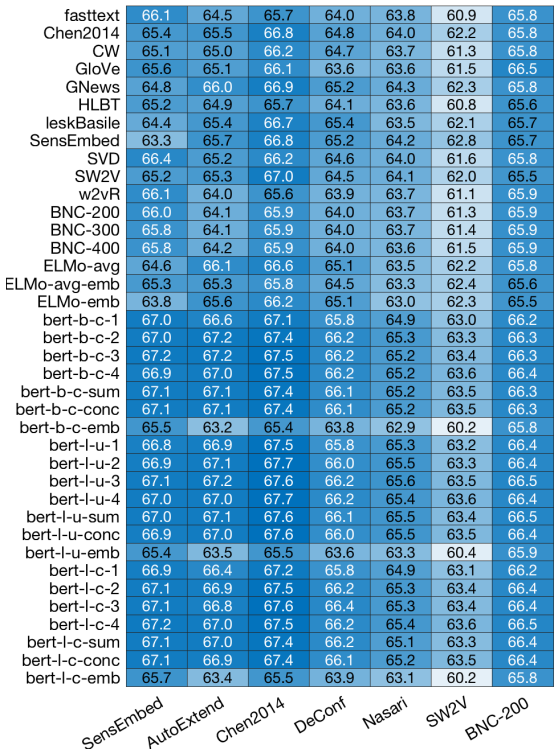

Figure 2: Performances of the model on the union of all datasets. The results are presented as F1 for all combinations of word and sense embeddings. Word vectors are on the rows and sense vectors on the columns.

Contextualized word embeddings As contextualized embeddings we used ELMo (Peters et al., 2018) in three different configurations, namely: ELMo-avg, weighted sum of its three layers; ELMo-avg emb, weighted sum of its three layers and the embeddings it produces; and ELMo-emb, the word embeddings produced by the model[1]. We also tested three implementations of BERT (Devlin et al., 2019): base cased (b-c); large uncased (l-u) and large cased (l-c). They offer pre-trained deep bidirectional representations of words in context[2]. We used seven configurations for each model: one for each of the last four layers (numbered from 1 to 4), the sum of these layers, their concatenation and the embedding layer. We fed all these models with the entire texts of the datasets. Since BERT uses WordPiece tokenization, we averaged sub-token embeddings to obtain token-level representations.

We also included three models which were built together with the sense embeddings introduced below.

Sense embeddings As sense embeddings, in addition to the three models introduced in Section 3 (AutoExtend, NASARI and SensEmbed), we included four models: Chen et al. (2014), a unified model which learns sense vectors by training a sense-annotated corpus disambiguated with a framework based on semantic similarity of WordNet sense definitions; meanBNC, created using a weighted combination of the words from WordNet glosses, using, as word vectors, the set of BNC-200 mentioned earlier; DeConf (Pilehvar and Collier, 2016), also linked to WordNet, a model where sense vectors are inferred in the same semantic space of pre-trained word embeddings by decomposing the given word representation into its constituent senses; and finally SW2V (Mancini et al., 2017), a model linked to BabelNet which uses a shallow disambiguation step and which, by extending the word2vec architecture, learns word and sense vectors jointly in the same semantic space as an emerging feature.

Results The results of these models are reported in Figure 2. One of the most interesting patterns that emerges from the heat map is that there are some combinations of word and sense embeddings that always work better than others. Sense vectors drive the performance of the system, contributing in great part to the accumulation of payoffs during the games. The sense vectors that maintain high performances are SensEmbed, AutoExtended and Chen2014. In particular Chen2014 has high performances with all the word embedding combinations. While these models are specific sense embedding techniques, the construction of BNC-200 follows a very simple method, which in view of these results can be refined using more principled gloss embedding techniques. The performances of

[1]TensorFlow models available at https://tfhub.dev/google/elmo/2.

[2]PyTorch models available at https://github.com/huggingface/pytorch-transformers.

	model	S2	S3	SE07	SE13	SE15	All	N	V	A	R
semi-sup.	*MFS*	64.7*	65.4	53.9	62.9	66.6*	64.1	68.1	49.5	74.1	80.6
	Babelfy	67.0	63.5*	51.6*	66.4†	70.3	65.5*	68.6*	49.9	73.2	79.8
	ppr_w2w	68.8	66.1	53.0	**68.8**	70.3	67.3	-	-	-	-
	WSD-TM	69.0	66.9	55.6	65.3*	69.6	66.9	69.7†	51.2	**76.0**	**80.9**
	$WSDG_{\alpha}$	68.7	**68.3**	**58.9**	66.4	70.7	**67.7**	**71.1**	**51.9**†	75.4	**80.9**
	$WSDG_{\beta}$	68.9	65.5	54.5	67.0	**72.8**	67.2	70.4	51.3	75.7	80.6
	$WSDG_{\gamma}$	**69.3**	66.4	56.0†	65.9	70.8	67.2	70.4	51.5	75.1	80.6
sup.	*IMS (2010)*	70.9†	69.3	61.3	65.3	69.5†	68.9†	70.5	55.8	75.6	82.9
	IMS_{w2v}	72.2	70.4	62.6	65.9	71.5	70.1	71.9	56.6	75.9	84.7
	$Yuan_{LSTM}$	73.8	71.8	63.5	69.5	72.6	71.5	-	-	-	-
	$Raganato_{BLSTM}$	72.0	69.1	64.8	66.9	71.5	69.9	71.5	57.5	75.0	83.8
	GAS	72.2	70.5†	-	67.2	72.6	-	-	-	-	-
	fastSense	73.5	73.5	62.4	66.2	73.2	-	-	-	-	-

Table 1: Comparison with state-of-the-art algorithms: unsupervised or knowledge-based (*unsup.*), and supervised (*sup.*). *MFS* refers to the MFS heuristic computed on SemCor on each dataset. The results are provided as F1 and the first result of the semi supervised systems with a statistically significant difference from the best of each dataset is marked with * (χ^2, $p < 0.1$). † indicates the same statistics but including also supervised models.

NASARI are lower compared to lexical vectors: this may be due to our choice to use NASARI-embed, whose vectors have low dimensionality.

The word vectors that have consistently high performances in association with the three sense vectors mentioned above are BERT, Chen2014, SensEmbed and SW2V. This is not surprising since they are able to produce contextualised word representations, performing, in fact, a preliminary disambiguation of the words. In particular, SW2V is specifically tailored for WSD. ELMo and fastText perform slightly worse. The vectors constructed using syntactic information and trained on the BNC corpus have similar performances to the their counterparts trained on larger corpora without the use of syntactic information. If we focus on BERT, we can see that it is able to maintain high performances ($F1 \approx 67$) with all its configurations, except for the embedding layers of all the models (*-emb). The contribution of the sum and concatenation operations is not significant.

Comparison We performed a comparison with 3 configurations of our model, one for each of the three best sense vectors: $WSDG_{\alpha}$, obtained using Chen2014 as sense vectors and BERT-l-u-4 as word vectors; $WSDG_{\beta}$, obtained using SensEmbed as sense vectors and BERT-l-c-4 as word vectors; and $WSDG_{\gamma}$, obtained using AutoExtended as sense vectors and BERT-l-u-3 as word vectors.

As comparison systems we included three semi-supervised approaches mentioned above, Babelfy (Moro et al., 2014), ppr_{w2w}, the best configuration of UKB (Agirre et al., 2018), and WSD-TM, introduced by Chaplot and Salakhutdinov (2018) (for this model we did not have the possibility to verify the results since its code is not available). In addition, we also report the performances of relevant supervised models, namely: It Makes Sense (Zhong and Ng, 2010, IMS), Iacobacci et al. (2016), Yuan et al. (2016), Raganato et al. (2017), Joulin et al. (2017) and Uslu et al. (2018).

The results of our evaluation are shown in Table 1. As we can see our model achieves state-of-the-art performances on four datasets and on S13 and S15 it performs better than many supervised systems. In general the gap between supervised and semi-supervised systems is reducing. This encourages new research in this direction. Our model fares particularly well on the disambiguation of nouns and verbs. However, the main gap between our models and supervised systems relies upon the disambiguation of verbs.

7 Analysis

Polysemy As expected, most of the errors made by $WSDG_{\alpha}$ are on highly polysemous words. Figure 3 shows that the number of wrong answers increases as the number of senses grows, and that the number of wrong answers starts to be higher than that of correct answers when the number of senses for a target word is in the range of 10-15 senses. The words with the highest number of errors are polysemous verbs such as: *say* (34), *make* (24), *find* (21), *have*, (17), *take* (15), *get*, (15) and *do* (13). These are words that in many NLP applications (especially those based on distributional models) are treated as stopwords.

Sense rank Mancini et al. (2017) show that senses which are not the most frequent ones are particularly challenging and most sense-based approaches fail to represent them properly. In Figure 4 we report the results of WSDG$_\alpha$ divided per sense rank, where it is possible to see how the performances of the system deteriorate as the rank of the correct sense increases. It is interesting to see that, in the first part of the plot, the performances follow a regular pattern that resembles a power-law distribution. This requires further analysis beyond the scope of this work, along the lines of Ferrer-i Cancho and Vitevitch (2018).

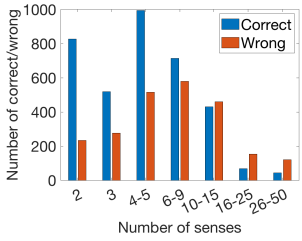

Figure 3: Correct and wrong answers given by WSDG$_\alpha$ grouped by number of senses

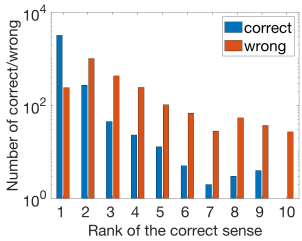

Figure 4: Correct and wrong answers given by WSDG$_\alpha$ per sense rank.

Priors Corroborating the findings of Pilehvar and Navigli (2014), Postma et al. (2016) conducted a series of experiments to study the effect that the variation of sense distributions in the training set has on the performances of *It makes sense* (Zhong and Ng, 2010). Specifically, they increased the volume of training examples (V) by enriching SemCor with senses inferred from BabelNet; increased the number of least frequent senses (LFS) (V+LFS); and overfitted the model constructing a training set proportional to the correct sense distribution of the test set (GOLD,GOLD+LFS). We used the same training sets to compute the priors for our system. The results of this analysis are reported in Table 2. These experiments show that increasing the number of training examples has a small beneficial effect. Increasing the number of LFS examples leads to worse results because this is a deviation from a real sense distribution. Further, to work with better semantic representations, this operation should also be accompanied by a similar selection on the training set of sense and word embeddings, otherwise LFS remain underrepresented. Finally, mimicking the distribution of the test set is more beneficial for *WSDG*$_\alpha$ than for IMS, especially when LFS examples are added, suggesting that semi-supervised systems can better adapt to specific domains than supervised systems.

System	V	V+LFS	GOLD	GOLD+LFS
IMS	68.9	62.0	86.8	85.4
WSDG$_\alpha$	66.4	57.5	88.4	90.8

Table 2: Comparison using different priors.

8 Exploratory study

We now present three WSD applications in as many tasks: selection of context-sensitive embeddings; sentence similarity; paraphrases detection.

Context-sensitive embeddings We used the Word in Context (WiC) dataset (Pilehvar and Camacho-Collados, 2019) for this task. It contains 7466 sentence pairs in which a target word appears in two different contexts. The task consisted of predicting if a target word has the same sense in the two sentences or not. The aim of this experiment was twofold: we wanted to show the usefulness of contextualized word embeddings obtained from WSD systems and to demonstrate that the model was able to maintain the textual coherence. The experiments on this dataset were conducted on the development set (1400 sentence pairs). The comparison was conducted against state-of-the-art models for contextualized word embeddings and sense embeddings: Context2Vec (Melamud et al., 2016) based on a bidirectional LSTM language model; ELMo$_1$, the first LSTM hidden state; ELMo$_3$, the weighted sum of the 3 LSTM layers; BERT$_{\text{base}}$; BERT$_{\text{large}}$. The results of these systems were taken from Pilehvar and Camacho-Collados (2019). We note here that all these models, including WSDG$_\alpha$, do not use training data. They are based on a simple threshold-based classifier, tuned on the development set (638 sentence pairs). WSDG$_\alpha$ disambiguates all the words in each pair of sentences separately and, if the cosine

C2V	$ELMo_1$	$ELMo_3$	$BERT_{base}$	$BERT_{large}$	$WSDG_\alpha$
59.7	57.1	56.3	63.6	63.8	**66.2**

Table 3: Performance on the WiC dataset.

	Pearson	Spearman	MSE
sense	46.5	43.9	7.9
word	39.8	39.9	8.6

Table 4: $WSDG_\alpha$ results on the SICK dataset.

similarity among the senses assigned to the target words is below a threshold (0.9), it classifies the pair as different senses, and as the same sense otherwise. As shown in Table 3 the disambiguation step has a big impact on the results.

Sentence similarity We used the SICK dataset (Marelli et al., 2014) for this task. It consists of 9841 sentence pairs that had been annotated with relatedness scores on a 5-point rating scale. We used the test split of this dataset that contains 4906 sentence pairs. The aim of this experiment was to test if disambiguated sense vectors can provide a better representation of sentences than word vectors. We used a simple method to test the two representations: it consisted of representing a sentence as the sum of the disambiguated sense vectors in one case and as the sum of word vectors in the other case. Once the sentence representations had been obtained for both methods the cosine similarity was used to measure their relatedness. The results of this experiment are reported in Table 4 as Pearson and Spearman correlation and Mean Squared Error (MSE). We used the α configuration of our model with Chen2014 to represent senses and BERT-l-u-4 to represent words. As we can see the simplicity of the method leads to low performances for both representations, but sense vectors correlate better than word vectors.

Paraphrase detection We used the test set of the Microsoft Research Paraphrase Corpus (Dolan et al., 2004, MRPC) for this task. The corpus contains 1621 sentence pairs that have been annotated with a binary label: 1 if the two sentences constitute a paraphrase and 0 otherwise. In this task we also used the sum of word vectors and the sum of disambiguated sense vectors to represent the sentences, and used part of the training set (10%) in order to tune the threshold parameter below which the sentences are not considered paraphrase. The classification accuracy for the word vectors used by $WSDG_\alpha$ was 58.1 whereas the disambiguated sense vectors obtained 66.9.

9 Conclusion

In this work we have presented *WSDG*, a flexible game-theoretic model for WSD. It combines game dynamics with most successful word and sense embeddings, therefore showing the potential of an effective combination of the two areas of game theory and word sense disambiguation.

Our approach achieves state-of-the-art performances on four datasets performing particularly well on the disambiguation of nouns and verbs. Beyond the numerical results, in this paper we have presented a model able to construct and evaluate word and sense representations. This is particularly useful since it can serve as a test bed for new word and sense embeddings. In particular, it will be interesting to test new sense embedding models based on contextual embeddings.

Thanks to the flexibility and scalability of our model, as future work we plan to explore in depth its use in different tasks, such as the creation of sentence (document) embeddings and lexical substitution. In fact, we believe that using disambiguated sense vectors, as shown in the context-sensitive embeddings and paraphrase detection studies, can offer a more accurate representation and improve the quality of downstream applications such as sentiment analysis and text classification (see, e.g., (Pilehvar et al., 2017)), machine translation and topic modelling. Encouraged by the good results achieved in our exploratory studies, we plan to develop a new model for contextualised word embeddings based on a game-theoretic framework.

Acknowledgments

The authors gratefully acknowledge the support of the ODYCCEUS project No. 732942 (first author) and of the ERC Consolidator Grant MOUSSE No. 726487 (second author) under the European Union's Horizon 2020 research and innovation programme.

The experiments have been run on the SCSCF cluster of Ca' Foscari University. The authors thank Ignacio Iacobacci for preliminary work on this paper.

References

Eneko Agirre, Oier López de Lacalle, and Aitor Soroa. 2018. The risk of sub-optimal use of open source NLP software: UKB is inadvertently state-of-the-art in knowledge-based WSD. In *Proceedings of Workshop for NLP Open Source Software (NLP-OSS)*, pages 29–33, Melbourne, Australia. ACL.

Eneko Agirre, Oier López de Lacalle, and Aitor Soroa. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1):57–84.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representations*.

Marco Baroni and Alessandro Lenci. 2010. Distributional memory: A general framework for corpus-based semantics. *Computational Linguistics*, 36(4):673–721.

Pierpaolo Basile, Annalina Caputo, and Giovanni Semeraro. 2014. An enhanced Lesk word sense disambiguation algorithm through a distributional semantic model. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 1591–1600, Dublin, Ireland. Dublin City University and ACL.

David M. Blei, Andrew Y. Ng, and Michael I. Jordan. 2003. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. NASARI: a novel approach to a semantically-aware representation of items. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 567–577, Denver, Colorado. ACL.

Ramon Ferrer-i Cancho and Michael S. Vitevitch. 2018. The origins of zipf's meaning-frequency law. *Journal of the Association for Information Science and Technology*, 69(11):1369–1379.

Devendra Singh Chaplot and Ruslan Salakhutdinov. 2018. Knowledge-based word sense disambiguation using topic models. In *AAAI Conference on Artificial Intelligence*.

Xinxiong Chen, Zhiyuan Liu, and Maosong Sun. 2014. A unified model for word sense representation and disambiguation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1025–1035, Doha, Qatar. ACL.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D. Manning. 2019. What does BERT look at? an analysis of BERT's attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 276–286, Florence, Italy. ACL.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. ICML*, pages 160–167.

Diego De Cao, Roberto Basili, Matteo Luciani, Francesco Mesiano, and Riccardo Rossi. 2010. Robust and efficient page rank for word sense disambiguation. In *Proceedings of TextGraphs-5 - 2010 Workshop on Graph-based Methods for Natural Language Processing*, pages 24–32, Uppsala, Sweden. ACL.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. ACL.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 350–356, Geneva, Switzerland. COLING.

Manaal Faruqui, Jesse Dodge, Sujay Kumar Jauhar, Chris Dyer, Eduard Hovy, and Noah A. Smith. 2015. Retrofitting word vectors to semantic lexicons. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1606–1615, Denver, Colorado. ACL.

Olivier Ferret and Brigitte Grau. 2002. A bootstrapping approach for robust topic analysis. *Nat. Lang. Eng.*, 8(3):209–233.

Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. 2013. PPDB: The paraphrase database. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 758–764, Atlanta, Georgia. ACL.

Taher H. Haveliwala. 2002. Topic-sensitive pagerank. In *Proceedings of the 11th International Conference on World Wide Web*, WWW '02, pages 517–526, New York, NY, USA. ACM.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2015. SensEmbed: Learning sense embeddings for word and relational similarity. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the*

7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 95–105, Beijing, China. ACL.

Ignacio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. 2016. Embeddings for word sense disambiguation: An evaluation study. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 897–907, Berlin, Germany. ACL.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. ACL.

Doo Soon Kim, Ken Barker, and Bruce Porter. 2010. Improving the quality of text understanding by delaying ambiguity resolution. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 581–589, Beijing, China. Coling 2010 Organizing Committee.

Minh Le, Marten Postma, Jacopo Urbani, and Piek Vossen. 2018. A deep dive into word sense disambiguation with LSTM. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 354–365, Santa Fe, New Mexico, USA. ACL.

Jiwei Li and Dan Jurafsky. 2015. Do multi-sense embeddings improve natural language understanding? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1722–1732, Lisbon, Portugal. ACL.

Fuli Luo, Tianyu Liu, Qiaolin Xia, Baobao Chang, and Zhifang Sui. 2018. Incorporating glosses into neural word sense disambiguation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia. ACL.

Massimiliano Mancini, Jose Camacho-Collados, Ignacio Iacobacci, and Roberto Navigli. 2017. Embedding words and senses together via joint knowledge-enhanced training. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 100–111, Vancouver, Canada. ACL.

Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. 2014. A SICK cure for the evaluation of compositional distributional semantic models. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*, pages 216–223, Reykjavik, Iceland. European Languages Resources Association (ELRA).

Oren Melamud, Jacob Goldberger, and Ido Dagan. 2016. context2vec: Learning generic context embedding with bidirectional LSTM. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 51–61, Berlin, Germany. ACL.

Rada Mihalcea, Paul Tarau, and Elizabeth Figa. 2004. PageRank on semantic networks, with application to word sense disambiguation. In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 1126–1132, Geneva, Switzerland. COLING.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *CoRR*, abs/1309.4168.

George A Miller, Claudia Leacock, Randee Tengi, and Ross T Bunker. 1993. A semantic concordance. In *Proceedings of the workshop on Human Language Technology*, pages 303–308. ACL.

Andriy Mnih and Geoffrey Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 641–648, New York, NY, USA. ACM.

Andrea Moro, Alessandro Raganato, and Roberto Navigli. 2014. Entity linking meets word sense disambiguation: a unified approach. *Transactions of the Association for Computational Linguistics*, 2:231–244.

John Nash. 1951. Non-cooperative games. *Annals of Mathematics*, 54(2):286–295.

Roberto Navigli. 2009. Word sense disambiguation: A survey. *ACM Comput. Surv.*, 41(2):10:1–10:69.

Roberto Navigli. 2018. Natural Language Understanding: Instructions for (present and future) use. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI 2018)*, pages 5697–5702.

Roberto Navigli and Mirella Lapata. 2010. An experimental study of graph connectivity for unsupervised word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(4):678–692.

Roberto Navigli and Simone Paolo Ponzetto. 2012. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artif. Intell.*, 193:217–250.

John von Neuman and Oskar Morgenstern. 1944. *Theory of games and economic behavior*. Princeton University Press.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab. Previous number = SIDL-WP-1999-0120.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. ACL.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. ACL.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2019. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, Minneapolis, Minnesota. ACL.

Mohammad Taher Pilehvar and Nigel Collier. 2016. De-conflated semantic representations. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1680–1690, Austin, Texas. ACL.

Mohammad Taher Pilehvar and Roberto Navigli. 2014. A large-scale pseudoword-based evaluation framework for state-of-the-art word sense disambiguation. *Computational Linguistics*, 40(4):837–881.

Mohammed Taher Pilehvar, José Camacho-Collados, Roberto Navigli, and Nigel Collier. 2017. Towards a Seamless Integration of Word Senses into Downstream NLP Applications. In *Proc. of the 55th Annual Meeting of the Association for Computational Linguistics (ACL 2017)*, pages 1857–1869, Vancouver, Canada. ACL.

Marten Postma, Ruben Izquierdo Bevia, and Piek Vossen. 2016. More is not always better: balancing sense distributions for all-words word sense disambiguation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3496–3506, Osaka, Japan. The COLING 2016 Organizing Committee.

Alessandro Raganato, Jose Camacho-Collados, and Roberto Navigli. 2017. 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*, pages 99–110, Valencia, Spain. ACL.

Sascha Rothe and Hinrich Schütze. 2015. AutoExtend: Extending word embeddings to embeddings for synsets and lexemes. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1793–1803, Beijing, China. ACL.

Peter D. Taylor and Leo B. Jonker. 1978. Evolutionary stable strategies and game dynamics. *Mathematical Biosciences*, 40(1):145 – 156.

Rocco Tripodi and Marcello Pelillo. 2017. A game-theoretic approach to word sense disambiguation. *Computational Linguistics*, 43(1):31–70.

Tolga Uslu, Alexander Mehler, Daniel Baumartz, and Wahed Hemati. 2018. FastSense: An efficient word sense disambiguation classifier. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan. ELRA.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Jörgen W. Weibull. 1997. *Evolutionary game theory*. MIT press.

Dayu Yuan, Julian Richardson, Ryan Doherty, Colin Evans, and Eric Altendorf. 2016. Semi-supervised word sense disambiguation with neural models. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1374–1385, Osaka, Japan. The COLING 2016 Organizing Committee.

Zhi Zhong and Hwee Tou Ng. 2010. It makes sense: A wide-coverage word sense disambiguation system for free text. In *Proceedings of the ACL 2010 System Demonstrations*, pages 78–83, Uppsala, Sweden. ACL.