A Game-Theoretic Approach to Word Sense Disambiguation

Rocco Tripodi[*]
Ca' Foscari University of Venice

Marcello Pelillo[**]
Ca' Foscari University of Venice

This article presents a new model for word sense disambiguation formulated in terms of evolutionary game theory, where each word to be disambiguated is represented as a node on a graph whose edges represent word relations and senses are represented as classes. The words simultaneously update their class membership preferences according to the senses that neighboring words are likely to choose. We use distributional information to weigh the influence that each word has on the decisions of the others and semantic similarity information to measure the strength of compatibility among the choices. With this information we can formulate the word sense disambiguation problem as a constraint satisfaction problem and solve it using tools derived from game theory, maintaining the textual coherence. The model is based on two ideas: Similar words should be assigned to similar classes and the meaning of a word does not depend on all the words in a text but just on some of them. The article provides an in-depth motivation of the idea of modeling the word sense disambiguation problem in terms of game theory, which is illustrated by an example. The conclusion presents an extensive analysis on the combination of similarity measures to use in the framework and a comparison with state-of-the-art systems. The results show that our model outperforms state-of-the-art algorithms and can be applied to different tasks and in different scenarios.

1. Introduction

Word Sense Disambiguation (WSD) is the task of identifying the intended meaning of a word based on the context in which it appears (Navigli 2009). It has been studied since the beginnings of Natural Language Processing (NLP) (Weaver 1955) and today it is still a central topic of this discipline. This because it is important for many NLP tasks such as text understanding (Kilgarriff 1997), text entailment (Dagan and Glickman 2004), machine translation (Vickrey et al. 2005), opinion mining (Smrž 2006), sentiment

[*] European Centre for Living Technology, Ca' Minich, S. Marco 2940 30124 Venezia, Italy.
E-mail: `rocco.tripodi@unive.it`.

[**] Dipartimento di Scienze Ambientali, Informatica e Statistica, Via Torino 155 - 30172 Venezia, Italy.
E-mail: `pelillo@unive.it`.

Submission received: 8 June 2015; revised submission received: 25 January 2016; accepted for publication: 24 March 2016.

doi:10.1162/COLI_a_00274

analysis (Rentoumi et al. 2009), and information extraction (Zhong and Ng 2012). All these applications can benefit from the disambiguation of ambiguous words, as a preliminary process; otherwise they remain on the surface of the word, compromising the coherence of the data to be analyzed (Pantel and Lin 2002).

To solve this problem, over the past few years, the research community has proposed several algorithms based on supervised (Tratz et al. 2007; Zhong and Ng 2010), semi-supervised (Navigli and Velardi 2005; Pham, Ng, and Lee 2005), and unsupervised (Mihalcea 2005; McCarthy et al. 2007) learning models. Nowadays, although supervised methods perform better in general domains, unsupervised and semi-supervised models are receiving increasing attention from the research community, with performances close to the state of the art of supervised systems (Ponzetto and Navigli 2010). In particular, knowledge-based and graph-based algorithms are emerging as promising approaches to solve the disambiguation problem (Sinha and Mihalcea 2007; Agirre et al. 2009). The peculiarities of these algorithms are that they do not require any corpus evidence and use only the structural properties of a lexical database to perform the disambiguation task. In fact, unsupervised methods are able to overcome a common problem in supervised learning—the knowledge acquisition problem, which requires the production of large-scale resources, manually annotated with word senses.

Knowledge-based approaches exploit the information from knowledge resources such as dictionaries, thesauri, or ontologies and compute sense similarity scores to disambiguate words in context (Mihalcea 2006). Graph-based approaches model the relations among words and senses in a text with graphs, representing words and senses as nodes and the relations among them as edges. From this representation the structural properties of the graph can be extracted and the most relevant concepts in the network can be computed (Agirre et al. 2006; Navigli and Lapata 2007).

Our approach falls within these two lines of research; it uses a graph structure to model the *geometry* of the data points (the words in a text) and a knowledge base to extract the senses of each word and to compute the similarity among them. The most important difference between our approach and state-of-the-art graph-based approaches (Véronis 2004; Sinha and Mihalcea 2007; Navigli and Lapata 2010; Agirre, de Lacalle, and Soroa 2014; Moro, Raganato, and Navigli 2014) is that in our method the graph contains only words and not senses. This graph is used to model the pairwise interaction among words and not to rank the senses in the graph according to their relative importance.

The starting point of our research is based on two fundamental assumptions:

1. The meaning of a sentence emerges from the interaction of the components that are involved in it.

2. These interactions are different and must be weighted in order to supply the right amount of information.

We interpret language as a complex adaptive system, composed of linguistic units and their interactions (Larsen-Freeman and Cameron 2008; Cong and Liu 2014). The interactions among units give rise to the emergence of properties, which, in our case, by problem definition, can be interpreted as meanings. In our model the relations between the words are weighted by a similarity measure with a distributional approach, increasing the weights among words that share a proximity relation. Weighting the interaction of the nodes in the graph is helpful in situations in which the indiscriminate use of contextual information can deceive. Furthermore, it models the idea that the meaning of a word does not depend on all the words in a text but just on some of them (Chaplot, Bhattacharyya, and Paranjape 2015).

This problem is illustrated in these sentences:

- There is a financial institution near the river bank.
- They were troubled by insects while playing cricket.

In these two sentences[1] the meaning of the words *bank* and *cricket* can be misinterpreted by a centrality algorithm that tries to find the most important node in the graph composed of all the possible senses of the words in the sentence. This because the meanings of the words *financial* and *institution* tend to shift the meaning of the word *bank* toward its financial meaning and not toward its naturalistic meaning. The same behavior can be observed for the word *cricket*, which is shifted by the word *insect* toward its insect meaning and not toward its game meaning. In our work, the disambiguation task is performed imposing a stronger importance on the relations between the words *bank* and *river* for the first sentence and between *cricket* and *play* for the second; exploiting proximity relations.

Our approach is based on the principle that the senses of the words that share a strong relation must be similar. The idea of assigning a similar class to similar objects has been implemented in a different way by Kleinberg and Tardos (2002), within a Markov random field framework. They have shown that it is beneficial in combinatorial optimization problems. In our case, this idea can preserve the textual coherence—a characteristic that is missing in many state-of-the-art systems. In particular, it is missing in systems in which the words are disambiguated independently. On the contrary, our approach disambiguates all the words in a text concurrently, using an underlying structure of interconnected links, which models the interdependence between the words. In so doing, we model the idea that the meaning for any word depends at least implicitly on the combined meaning of all the interacting words.

In our study, we model these interactions by developing a system in which it is possible to map lexical items onto concepts exploiting contextual information in a way in which collocated words influence each other simultaneously, imposing constraints in order to preserve the textual coherence. For this reason, we have decided to use a powerful tool, derived from game theory: the non-cooperative game (see Section 4). In our system, the nodes of the graph are interpreted as players, in the game theoretic sense (see Section 4), that play a game with the other words in the graph in order to maximize their utility; constraints are defined as similarity measures among the senses of two words that are playing a game. The concept of utility has been used in different ways in the game theory literature; in general, it refers to the satisfaction that a player derives from the outcome of a game (Szabó and Fath 2007). From our point of view, increasing the utility of a word means increasing the textual coherence in a distributional semantics perspective (Firth 1957). In fact, it has been shown that collocated words tend to have a determined meaning (Gale, Church, and Yarowsky 1992; Yarowsky 1993).

Game theoretic frameworks have been used in different ways to study language use (Pietarinen 2007; Skyrms 2010) and evolution (Nowak, Komarova, and Niyogi 2001), but to the best of our knowledge, our method is the first attempt to use it in a specific NLP task. This choice is motivated by the fact that game theoretic models are able to perform a consistent labeling of the data (Hummel and Zucker 1983; Pelillo

[1] A complete example of the disambiguation of the first sentence is given in Section 5.3.

1997), taking into account contextual information. These features are of great importance for an unsupervised or semi-supervised algorithm, which tries to perform a WSD task, because, by assumption, the sense of a word is given by the context in which it appears. Within a game theoretic framework we are able to cast the WSD problem as a continuous optimization problem, exploiting contextual information in a dynamic way. Furthermore, no supervision is required and the system can adapt easily to different contextual domains, which is exactly what is required for a WSD algorithm.

The additional reason for the use of a consistent labeling system relies on the fact that it is able to deal with *semantic drifts* (Curran, Murphy, and Scholz 2007). In fact, as shown in the two example sentences, concentrating the disambiguation task of a word on highly collocated words, taking into account proximity (or even syntactic) information, allows the meaning interpretation to be guided only towards senses that are strongly related to the word that has to be disambiguated.

In this article, we provide a detailed discussion about the motivation behind our approach and a full evaluation of our algorithm, comparing it with state-of-the-art systems in WSD tasks. In a previous work we used a similar algorithm in a semi-supervised scenario (Tripodi, Pelillo, and Delmonte 2015), casting the WSD task as a graph transduction problem. Now we have extended that work, making the algorithm fully unsupervised. Furthermore, in this article we provide a complete evaluation of the algorithm extending our previous works (Tripodi and Pelillo 2015), exploiting proximity relations among words.

An important feature of our approach is that it is versatile. In fact, the method can adapt to different scenarios and to different tasks, and it is possible to use it as unsupervised or semi-supervised. The semi-supervised approach, presented in Tripodi, Pelillo, and Delmonte (2015), is a bootstrapping graph-based method, which propagates, over the graph, the information from labeled nodes to unlabeled. In this article, we also provide a new semi-supervised version of the approach, which can exploit the evidence from sense tagged corpora or the most frequent sense heuristic and does not require labeled nodes to propagate the labeling information.

We tested our approach on different data sets from WSD and entity-linking tasks in order to find the similarity measures that perform better, and evaluated our approach against unsupervised, semi-supervised, and supervised state-of-the-art systems. The results of this evaluation show that our method performs well and can be considered as a valid alternative to current models.

2. Related Work

There are two major paradigms in WSD: supervised and knowledge-based. Supervised algorithms learn, from sense-labeled corpora, a computational model of the words of interest. Then, the obtained model is used to classify new instances of the same words. Knowledge-based algorithms perform the disambiguation task by using an existing lexical knowledge base, which usually is structured as a semantic network. Then, these approaches use graph algorithms to disambiguate the words of interest, based on the relations that these words' senses have in the network (Pilehvar and Navigli 2014).

A popular supervised WSD system, which has shown good performances in different WSD tasks, is *It Makes Sense* (Zhong and Ng 2010). It takes as input a text and for each content word (noun, verb, adjective, or adverb) outputs a list of possible senses ranked according to the likelihood of appearing in a determined context and extracted from a knowledge base. The training data used by this system are derived from SemCor (Miller et al. 1993), DSO (Ng and Lee 1996), and collected automatically exploiting parallel corpora (Chan and Ng 2005). Its default classifier is LIBLINEAR[2] with a linear kernel and its default parameters.

Unsupervised and knowledge-based algorithms for WSD are attracting great attention from the research community. This is because supervised systems require training data, which are difficult to obtain. In fact, producing sense-tagged data is a time-consuming process, which has to be carried out separately for each language of interest. Furthermore, as investigated by Yarowsky and Florian (2002), the performance of a supervised algorithm degrades substantially with an increase of sense entropy. **Sense entropy** refers to the distribution over the possible senses of a word, as seen in training data. Additionally, a supervised system has difficulty in adapting to different contexts, because it depends on prior knowledge, which makes the algorithm rigid; therefore, it cannot efficiently adapt to domain specific cases, when other optimal solutions may be available (Yarowsky and Florian 2002).

One of the most common heuristics that allows us to exploit sense tagged data such as SemCor (Miller et al. 1993) is the most frequent sense. It exploits the overall sense distribution for each word to be disambiguated, choosing the sense with the highest probability regardless of any other information. This simple procedure is very powerful in general domains but cannot handle senses with a low distribution, which can be found in specific domains.

With these observations in mind, Koeling et al. (2005) created three domain-specific corpora to evaluate WSD systems. They tested whether WSD algorithms are able to adapt to different contexts, comparing their results with the most frequent sense heuristic computed on general domains corpora. They used an unsupervised approach to obtain the most frequent sense for a specific domain (McCarthy et al. 2007) and demonstrated that their approach outperforms the most frequent sense heuristic derived from general domain and labeled data.

This heuristics for the unsupervised acquisition of the predominant sense of a word consists of collecting all the possible senses of a word and then in ranking these senses. The ranking is computed according to the information derived from a distributional thesaurus automatically produced from a large corpus and a semantic similarity measure derived from the sense inventory. Although the authors have demonstrated that this approach is able to outperform the most frequent sense heuristic computed on sense-tagged data on general domains, it is not easy to use it on real world applications, especially when the domain of the text to be disambiguated is not known in advance.

Other unsupervised and semi-supervised approaches, rather than computing the prevalent sense of a word, try to identify the actual sense of a word in a determined phrase, exploiting the information derived from its context. This is the case with traditional algorithms, which exploit the pairwise semantic similarity among a target word and the words in its context (Lesk 1986; Resnik 1995; Patwardhan, Banerjee, and Pedersen 2003). Our work could be considered as a continuation of this tradition, which tries to identify the intended meaning of a word given its context, using a new approach for the computation of the sense combinations.

Graph-based algorithms for WSD are gaining much attention in the NLP community. This is because graph theory is a powerful tool that can be used both for the organization of the contextual information and for the computation of the relations among word senses. It allows us to extract the structural properties of a text. Examples of this kind of approach construct a graph from all the senses of the words in a text and

[2] `http://liblinear.bwaldvogel.de`.

then use connectivity measures in order to identify the most relevant word senses in the graph (Navigli and Lapata 2007; Sinha and Mihalcea 2007). Navigli and Lapata (2007) conducted an extensive analysis of graph connectivity measures for unsupervised WSD. Their approach uses a knowledge base, such as WordNet, to collect and organize all the possible senses of the words to be disambiguated in a graph structure, then uses the same resource to search for a path (of predefined length) between each pair of senses in the graph. Then, if it exists, it adds all the nodes and edges on this path to the graph. These measures analyze local and global properties of the graph. **Local measures**, such as degree centrality and eigenvector centrality, determine the degree of relevance of a single vertex. **Global properties**, such as compactness, graph entropy, and edge density, analyze the structure of the graph as a whole. The results of the study show that local measures outperform global measures and, in particular, that degree centrality and PageRank (Page et al. 1999) (which is a variant of the eigenvector centrality measure) achieve the best results.

PageRank (Page et al. 1999) is one of the most popular algorithms for WSD; in fact, it has been implemented in different ways by the research community (Haveliwala 2002; Mihalcea, Tarau, and Figa 2004; De Cao et al. 2010; Agirre, de Lacalle, and Soroa 2014). It represents the senses of the words in a text as nodes of a graph. It uses a knowledge base to collect the senses of the words in a text and represents them as nodes of a graph. The structure of this resource is used to connect each node with its related senses in a directed graph. The main idea of this algorithm is that whenever a link from a node to another exists, a vote is produced, increasing the rank of the voted node. It works by counting the number and quality of links to a node in order to determine an estimation of how important the node is in the network. The underlying assumption is that more important nodes are likely to receive more links from other nodes (Page et al. 1999). Exploiting this idea, the ranking of the nodes in the graph can be computed iteratively with the following equation:

$$Pr = cMPr + (1 - c)v \tag{1}$$

where M is the transition matrix of the graph, v is an $N \times 1$ vector representing a probability distribution, and c is the so-called damping factor that represents the chance that the process stops, restarting from a random node. At the end of the process each word is associated with the most important concept related to it. One problem of this framework is that the labeling process is not assumed to be consistent.

One algorithm that tries to improve centrality algorithms is SUDOKU, introduced by Minion and Sainudiin (2014). It is an iterative approach, which simultaneously constructs the graph and disambiguates the words using a centrality function. It starts inserting the nodes corresponding to the senses of the words with low polysemy and iteratively inserting the more ambiguous words. The advantages of this method are that the use of small graphs, at the beginning of the process, reduces the complexity of the problem and that it can be used with different centrality measures.

Recently, a new model for WSD has been introduced, based on an undirected graphical model (Chaplot, Bhattacharyya, and Paranjape 2015). It approaches the WSD problem as a maximum a posteriori query on a Markov random field (Jordan and Weiss 2002). The graph is constructed using the content words of a sentence as nodes and connecting them with edges if they share a relation, determined using a dependency parser. The values that each node in the graphical model can take include the senses of the corresponding word. The senses are collected using a knowledge base and weighted using a probability distribution based on the frequency of the senses in the knowledge base. Furthermore, the senses between two related words are weighted using a similarity measure. The goal of this approach is to maximize the joint probability of the senses of all the words in the sentence, given the dependency structure of the sentences, the frequency of the senses, and the similarity among them.

A new graph-based, semi-supervised approach introduced to deal with multilingual WSD (Navigli and Ponzetto 2012b) and entity inking problems is Babelfy (Moro, Raganato, and Navigli 2014). Multilingual WSD is an important task because traditional WSD algorithms and resources are focused on English language. It exploits the information from large multilingual knowledge, such as BabelNet (Navigli and Ponzetto 2012a), to perform this task. Entity linking consists of disambiguating the named entities in a text and in finding the appropriate resources in an ontology, which correspond to the specific entities mentioned in a text. Babelfy creates the semantic signature of each word to be disambiguated, which consists of collecting, from a semantic network, all the nodes related to a particular concept, exploiting the global structure of the network. This process leads to the construction of a graph-based representation of the whole text. It then applies Random Walk with Restart (Tong, Faloutsos, and Pan 2006) to find the most important nodes in the network, solving the WSD problem.

Approaches that are more similar to ours in the formulation of the problem have been described by Araujo (2007). The author reviewed the literature devoted to the application of different evolutionary algorithms to several aspects of NLP: syntactical analysis, grammar induction, machine translation, text summarization, semantic analysis, document clustering, and classification. Basically, these approaches are search and optimization methods inspired by biological evolution principles. A specific evolutionary approach for WSD has been introduced by Menai (2014). It uses genetic algorithms (Holland 1975) and memetic algorithms (Moscato 1989) in order to improve the performances of a *gloss-based* method. It assumes that there is a population of individuals, represented by all the senses of the words to be disambiguated, and that there is a selection process, which selects the best candidates in the population. The selection process is defined as a sense similarity function, which gives a higher score to candidates with specific features, increasing their *fitness* to the detriment of the other population members. This process is repeated until the *fitness* level of the population regularizes and at the end the candidates with higher *fitness* are selected as solutions of the problem. Another approach, which addresses the disambiguation problem in terms of space search, is GETALP (Schwab et al. 2013). This uses an Ant Colony algorithm to find the best path in the weighted graph constructed, measuring the similarity of all the senses in a text and assigning to each word to be disambiguated the sense corresponding to the node in this path.

These methods are similar to our study in the formulation of the problem; the main difference is that our approach is defined in terms of evolutionary game theory. As we show in the next section, this approach ensures that the final labeling of the data is consistent and that the solution of the problem is always found. In fact, our system always converges to the nearest Nash equilibrium from which the dynamics have been started.

3. Word Sense Disambiguation as a Consistent Labeling Problem

WSD can be interpreted as a sense-labeling task (Navigli 2009), which consists in assigning a sense label to a target word. As a labeling problem we need an algorithm, which performs this task in a consistent way, taking into account the context in which the target word occurs. Following this observation, we can formulate the WSD task as a constraint satisfaction problem (Tsang 1995) in which the labeling process has to satisfy some constraints in order to be consistent. This approach gives us the possibility not only to exploit the contextual information of a word but also to find the most appropriate sense association for the target word and the words in its context. This is the most important contribution of our work, which distinguishes it from existing WSD algorithms. In fact, in some cases using only contextual information without the imposition of constraints can lead to inconsistencies in the assignment of senses to related words.

As an illustrative example we can consider a binary constraint satisfaction problem, which is defined by a set of variables representing the elements of the problem and a set of binary constraints representing the relationships among variables. The problem is considered solved if there is a solution that satisfies all the constraints. This setting can be described in a formal manner as a triple (V, D, R), where $V = \{v_1, ..., v_n\}$ is the set of variables; $D = \{D_{v_1}, ..., D_{v_n}\}$ is the set of domains for each variable, each D_{v_i} denoting a finite set of possible values for variable v_i; and $R = \{R_{ij} | R_{ij} \subseteq D_{v_i} \times D_{v_j}\}$ is a set of binary constraints where R_{ij} describe a set of compatible pairs of values for the variables v_i and v_j. R can be defined as a binary matrix of size $p \times q$ where p and q are the cardinalities of domains and variables, respectively. Each element of the binary matrix $R_{ij}(\lambda, \lambda^{'}) = 1$ indicates if the assignment $v_i = \lambda$ is compatible with the assignment $v_j = \lambda^{'}$. R is used to impose constraints on the labeling so that each label assignment is consistent.

This binary case assumes that the constraints are completely violated or completely respected, which is restrictive; it is more appropriate, in many real-world cases, to have a weight that expresses the level of confidence about a particular assignment (Hummel and Zucker 1983). This notion of consistency has been shown to be related to the Nash equilibrium concept in game theory (Miller and Zucker 1991). We have adopted this method to approach the WSD task in order to perform a consistent labeling of the data. In our case, we can consider variables as words, labels as word senses, and compatibility coefficients as similarity values among two word senses. To explain how the Nash equilibria are computed we need to introduce basic notions of game theory in the following section.

4. Game Theory

In this section, we briefly introduce the basic concepts of classical game theory and evolutionary game theory that we used in our framework; for a more detailed analysis of these topics, the reader is referred to Weibull (1997), Leyton-Brown and Shoham (2008), and Sandholm (2010).

4.1 Classical Game Theory

Game theory provides predictive power in interactive decision situations. It was introduced by Von Neumann 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 (which is the one we use in this article) it consists of a finite set of players $I = \{1, .., n\}$, a set of pure strategies for each player $S_i = \{s_1, ..., s_m\}$, and a utility function $u_i : S_1 \times ... \times S_n \rightarrow \mathbb{R}$, which associates strategies to payoffs. Each player can adopt a strategy in order to play a game; and the utility function depends on the combination of strategies played at the same time by the players involved in the game, not just on the strategy chosen by a single player. An important assumption in game theory is that the players are rational and try to maximize the value of

u_i; Furthermore, in **non-cooperative games** the players choose their strategies independently, considering what the other players can play and try to find the best strategy profile to use in a game.

A strategy s_i^* is said to be *dominant* if and only if:

$$u_i(s_i^*, s_{-i}) > u_i(s_i, s_{-i}), \forall s_{-i} \in S_{-i} \tag{2}$$

where S_{-i} represents all strategy sets other than player i's.

As an example, we can consider the famous *Prisoner's Dilemma*, whose payoff matrix is shown in Table 1. Each cell of the matrix represents a strategy profile, where the first number represents the payoff of *Player 1* (P_1) and the second is the payoff of *Player 2* (P_2), when both players use the strategy associated with a specific cell. P_1 is called the **row player** because it selects its strategy according to the rows of the payoff matrix, and P_2 is called the **column player** because it selects its strategy according to the columns of the payoff matrix. In this game the strategy *confess* is a *dominant strategy* for both players and this strategy combination is the *Nash equilibrium* of the game.

Nash equilibria represent the key concept of game theory and can be defined as those strategy profiles in which each strategy is a best response to the strategy of the co-player and no player has the incentive to unilaterally deviate from their decision, because there is no way to do better.

In many games, the players can also play **mixed strategies**, which are probability distributions over their pure strategies. Within this setting, the players choose a strategy with a certain pre-assigned probability. A mixed strategy set can be defined as a vector $x = (x_1, \ldots, x_m)$, where m is the number of pure strategies and each component x_h denotes the probability that player i chooses its hth pure strategy. For each player, the strategy set is defined as a standard simplex:

$$\Delta = \left\{ x \in \mathbb{R}^n : \sum_{h=1}^{m} x_h = 1, \text{ and } x_h \geq 0 \text{ for all } h \in x \right\} \tag{3}$$

Each mixed strategy corresponds to a point on the simplex and its corners correspond to pure strategies.

In a *two-player game* we can define a strategy profile as a pair (p, q) where $p \in \Delta_i$ and $q \in \Delta_j$. The expected payoff for this strategy profile is computed as follows: $u_i(p, q) = p \cdot A_i q$ and $u_j(p, q) = q \cdot A_j p$, where A_i and A_j are the payoff matrices of player i and player j, respectively. The Nash equilibrium is computed in mixed strategies in the same way as pure strategies. It is represented by a pair of strategies such that each is a best response to the other. The only difference is that, in this setting, the strategies are probabilities and must be computed considering the payoff matrix of each player.

Table 1
The Prisoner's Dilemma.

$P_1 \backslash P_2$	confess	don't confess
confess	−5,−5	0,−6
don't confess	−6,0	−1,−1

A game theoretic framework can be considered as a solid tool in decision-making situations because a fundamental theorem by Nash (1951) states that any normal-form game has at least one mixed Nash equilibrium, which can be used as the solution of the decision problem.

4.2 Evolutionary Game Theory

Evolutionary game theory was introduced by Smith and Price (1973), overcoming some limitations of traditional game theory, such as the hyper-rationality imposed on the players. In fact, in real-life situations the players choose a strategy according to heuristics or social norms (Szabó and Fath 2007). It has been introduced in biology to explain the evolution of species. In this context, strategies correspond to phenotypes (traits or behaviors), payoffs correspond to offspring, allowing players with a high actual payoff (obtained thanks to their phenotype) to be more prevalent in the population. This formulation explains natural selection choices among alternative phenotypes based on their utility function. This aspect can be linked to rational choice theory, in which players make a choice that maximizes their utility, balancing cost against benefits (Okasha and Binmore 2012).

This intuition introduces an **inductive learning** process, in which we have a population of agents who play games repeatedly with their neighbors. The players at each iteration update their beliefs on the state of the game and choose their strategy according to what has been effective and what has not in previous games. The strategy space of each player i is defined as a mixed strategy profile x_i, as defined in the previous section, which lives in the mixed strategy space of the game, given by the Cartesian product:

$$\Theta = \times_{i \in I} \Delta_i \tag{4}$$

The expected payoff of a pure strategy e^h in a single game is calculated as in mixed strategies. The difference in evolutionary game theory is that a player can play the game with all other players, obtaining a final payoff, which is the sum of all the partial payoffs obtained during the single games. We have that the payoff relative to a single strategy is: $u_i(e_i^h) = \sum_{j=1}^{n}(A_{ij}x_j)_h$. The average payoff $u_i(x) = \sum_{j=1}^{n} x_i^T A_{ij} x_j$, where n is the number of players with whom the games are played and A_{ij} is the payoff matrix between players i and j. Another important characteristic of evolutionary game theory is that the games are played repeatedly. In fact, at each iteration a player can update their strategy space according to the payoffs gained during the games. They can allocate more probability to the strategies with high payoff until an equilibrium is reached. In order to find those states that correspond to the Nash equilibria of the games, the replicator dynamic equation is used (Taylor and Jonker 1978):

$$\dot{x} = [u(e^h, x) - u(x, x)] \cdot x^h \; \forall h \in x \tag{5}$$

which allows better than average strategies (best replies) to grow at each iteration.

The following theorem states that with Equation (5) it is always possible to find the Nash equilibria of the games (see Weibull [1997] for the proof).

Theorem 1
A point $x \in \Theta$ is the limit of a trajectory of Equation (5) starting from the interior of Θ if and only if x is a Nash equilibrium. Further, if point $x \in \Theta$ is a strict Nash equilibrium, then it is asymptotically stable, additionally implying that the trajectories starting from all nearby states converge to x.

As in Erdem and Pelillo (2012), we used the discrete time version of the replicator dynamic equation for the experiments of this article:

$$x^h(t+1) = x^h(t)\frac{u(e^h, x)}{u(x,x)} \; \forall h \in S \tag{6}$$

where, at each time step t, the players update their strategies according to the strategic environment until the system converges and the Nash equilibria are met. In classical evolutionary game theory these dynamics describe a stochastic evolutionary process in which the agents adapt their behaviors to the environment.

For example, if we analyze the Prisoner's Dilemma within the evolutionary game theory framework, we can see that the cooperative strategy (*do not confess*) tends to emerge as an equilibrium of the game and this is the best situation for both players, because this strategy gives a higher payoff than the defect strategy (*confess*), which is the equilibrium in the classical game theory framework. In fact, if the players play the game shown in Table 1 repeatedly and randomize their decisions in each game, assigning at the beginning a normal distribution to their strategies, their payoffs $u(x_{pi})$ can be computed as follows:

$$u(x_{p1}) = A_{p1}x_{p_2} = \begin{pmatrix} -5, & 0 \\ -6, & -1 \end{pmatrix} \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} = \begin{pmatrix} -2.5 \\ -3.5 \end{pmatrix}$$

$$u(x_{p2}) = A^T_{p2}x_{p_1} = \begin{pmatrix} -5, & -6 \\ 0, & -1 \end{pmatrix}^T \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} = \begin{pmatrix} -2.5 \\ -3.5 \end{pmatrix}$$

where T is the transpose operator required for P_2, which chooses its strategies according to the columns of the matrix in Table 1. This operation makes the matrices A_{p1} and A_{p2} identical and for this reason in this case the distinction among the two players is not required because they get the same payoffs. Now we can compute the strategy space of a player at time $t+1$ according to Equation (5):

$$x_1\colon -1.25/-3 = 0.42$$

$$x_2\colon -1.75/-3 = 0.58$$

The game is played with the new strategy spaces until the system converges—that is, when the difference among the payoffs at time t_n and t_{n-1} is under a small threshold. In Figure 1 we can see how the *cooperate strategy* increases over time, reaching a stationary point, which corresponds to the equilibrium of the game.

5. WSD Games

In this section we describe how the WSD games are formulated. We assume that each player $i \in I$ that participates in the games is a particular word in a text and that each strategy is a particular word sense. The players can choose a determined strategy

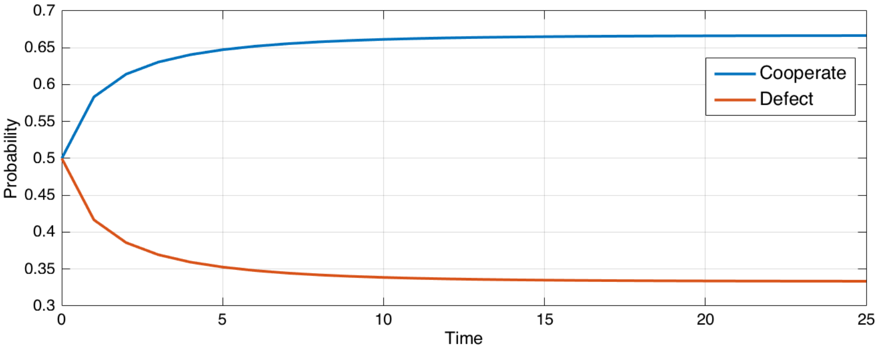


Figure 1
The dynamics of the repeated Prisoner's Dilemma.

among the set of strategies $S_i = \{1, ..., c\}$, each expressing a certain hypothesis about its membership in a class and c being the total number of classes available. We consider S_i as the mixed strategy for player i as described in Section 4. The games are played between two similar words, i and j, imposing only pairwise interaction between them. The payoff matrix Z_{ij} of a single game is defined as a sense similarity matrix between the senses of word i and word j. The payoff function for each word is additively separable and is computed as described in Section 4.2.

Formulating the problem in this way we can apply Equation (6) to compute the equilibrium state of the system, which corresponds to a consistent labeling of the data. In fact, once stability is reached, all players play the strategy with the highest payoff. Each player arrives at this state not only considering its own strategies but also the strategies that its co-players are playing. For each player $i \in I$ is chosen the strategy with the highest probability when the system converges (see Equation (7)).

$$\phi_i = \underset{h=1,\dots,c}{\arg\max}\ x_{ih} \tag{7}$$

In our framework, a word is not disambiguated only if it is not able to update its strategy space. This can happen when the player's strategy space is initialized with a uniform distribution and either its payoff matrices have only zero entries (i.e., when its senses are not similar to the senses of the co-players), or it is not connected with other nodes in the graph. The former assumption depends on the semantic measures used to calculate the payoffs (see Section 5.2.2); experimentally, we noticed that it does not happen frequently. The latter assumption can happen when a word is not present in a determined corpus. It can be avoided using query expansion techniques or connecting the disconnected node with nodes in its neighborhood, exploiting proximity relations (see Section 5.1.1). With Equation (7), it is guaranteed that at the end of the process each word is mapped to exactly one sense. Experimentally, we noticed that when a word is able to update its strategy space, it is not the case that two strategies in it have the same probability.

5.1 Implementation of the WSD Games

In order to run our algorithm, we need the network that models the interactions among the players, the strategy space of the game, and the payoff matrices. We adopted the following steps in order to model the data required by our framework and, specifically, for each text to be disambiguated we:

- Extract from the text the list of words I that have an entry in a lexical database
- compute from I the word similarity matrix W in which are stored the pairwise similarities among each word with the others and represents the players' interactions
- increase the weights between two words that share a proximity relation
- extract from I the list C of all the possible senses that represent the strategy space of the system
- assign for each word in I a probability distribution over the senses in C creating for each player a probability distribution over the possible strategies
- compute the sense similarity matrix Z among each pair of senses in C, which is then used to compute the partial payoff matrices of each game
- apply the replicator dynamics equation in order to compute the Nash equilibria of the games and
- assign to each word $i \in I$ a strategy $s \in C$

These steps are described in the following sections. In Section 5.1.1 we describe the graph construction procedure that we used in order to model the geometry of the data. In Section 5.1.2 we explain how we implement the strategy space of the game that allows each player to choose from a predetermined number of strategies. In Section 5.1.3 we describe how we compute the sense similarity matrix and how it is used to create the partial payoff matrices of the games. Finally, in Section 5.1.4 we describe the system dynamics.

5.1.1 Graph Construction. In our study, we modeled the geometry of the data as a graph. The nodes of the graph correspond to the words of a text, which have an entry in a lexical database. We denote the words by $I = \{i_j\}_{j=1}^{N}$, where i_j is the jth word and N is the total number of words retrieved. From I we construct an $N \times N$ similarity matrix W where each element w_{ij} is the similarity value assigned by a similarity function to the words i and j. W can be exploited as a useful tool for graph-based algorithms because it is treatable as a weighted adjacency matrix of a weighted graph.

A crucial factor for the graph construction is the choice of the similarity measure, $sim(\cdot,\cdot) \rightarrow \mathbb{R}$ to weight the edges of the graph. In our experiments, we used similarity measures, which compute the strength of co-occurrence between any two words i_i and i_j.

$$w_{ij} = sim(i_i, i_j)\ \forall i, j \in I : i \neq j \tag{8}$$

This choice is motivated by the fact that collocated words tend to have determined meanings (Gale, Church, and Yarowsky 1992; Yarowsky 1993), and also because the computation of these similarities can be obtained easily. In fact, it only required a corpus in order to compute a vast range of similarity measures. Furthermore, large corpora such as the BNC corpus (Leech 1992) and the Google Web 1T corpus (Brants and Franz 2006) are freely available and extensively used by the research community.

In some cases, it is possible that some target words are not present in the reference corpus, because of different text segmentation techniques or spelling differences. In this case, we use query expansion techniques in order to find an appropriate substitute (Carpineto and Romano 2012). Specifically, we use WordNet to find alternative lexicalizations of a lemma, choosing the one that co-occurs more frequently with the words in its context.

The information obtained from an association measure can be enriched by taking into account the proximity of the words in the text (or the syntactic structure of the sentence). The first task can be achieved augmenting the similarities among a target word and the n words that appear on its right and on its left, where n is a parameter that with small values can capture fixed expressions and with large values can detect semantic concepts (Fkih and Omri 2012). The second task can be achieved using a dependency parser to obtain the syntactical relations among the words in the target sentence, but this approach is not used in this article. In this way, the system is able to exploit local and global cues, mixing together the *one sense per discourse* (Kelly and Stone 1975) and the *one sense per collocation* (Yarowsky 1993) hypotheses.

We are not interested in all the relations in the sentence but we focus only on relations among target words. The use of a dependency/proximity structure makes the graph reflect the structure of the sentence and the use of a distributional approach allows us to exploit the relations of semantically correlated words. This is particularly useful when the proximity information is poor—for example, when it connects words to auxiliary or modal verbs. Furthermore, these operations ensure that there are no disconnected nodes in the graph.

5.1.2 Strategy Space Implementation. The strategy space of the game is created using a knowledge base to collect the sense inventories $M_i = \{1, \dots, m_i\}$ of each word in a text, where m_i is the number of senses associated with word i. Then we create the list $C = (1, \dots, c)$ of all the unique concepts in the sense inventories, which correspond to the space of the game.

With this information, we can define the strategy space S of the game in matrix form as:

$$\begin{array}{cccc} s_{i1} & s_{i2} & \cdots & s_{ic} \\ \vdots & \vdots & \cdots & \vdots \\ s_{n1} & s_{n2} & \cdots & s_{nc} \end{array}$$

where each row corresponds to the mixed strategy space of a player and each column corresponds to a specific sense. Each component s_{ih} denotes the probability that the player chooses to play its hth pure strategy among all the strategies in its strategy profile, as described in Section 4. The initialization of each mixed strategy space can either be uniform or take into account information from sense-labeled corpora.

5.1.3 The Payoff Matrices. We encoded the payoff matrix of a WSD game as a sense similarity matrix among all the senses in the strategy spaces of the game. In this way, the higher the similarity among the senses of two words, the higher the incentive for a word to choose that sense, and play the strategy associated with it.

The $c \times c$ sense similarity matrix Z is defined in Equation (9).

$$z_{ij} = ssim(s_i, s_j)\ \forall i, j \in C : i \neq j \tag{9}$$

This similarity matrix can be obtained using the information derived by the same knowledge base used to construct the strategy space of the game. It is used to extract the partial payoff matrix Z_{ij} for all the single games played between two players i and j. This operation is done extracting from Z the entries relative to the indices of the senses in the sense inventories M_i and M_j. It produces an $m_i \times m_j$ payoff matrix, where m_i and m_j are the numbers of senses in M_i and M_j, respectively.

5.1.4 System Dynamics. Now that we have the topology of the data W, the strategy space of the game S, and the payoff matrix Z, we can compute the Nash equilibria of the game according to Equation (6). In each iteration of the system, each player plays a game with its neighbors N_i according to the co-occurrence graph W. The payoffs of the hth strategy is calculated as:

$$u_i(e^h, x) = \sum_{j \in N_i} (w_{ij} Z_{ij} x_j)_h \tag{10}$$

and the player's payoff as:

$$u_i(x) = \sum_{j \in N_i} x_i^T (w_{ij} Z_{ij} x_j) \tag{11}$$

In this way we can weight the influence that each word has on the choices that a particular word has to make on its meaning. We assume that the payoff of word i depends on the similarity that it has with word j, w_{ij}, the similarities among its senses and those of word j, Z_{ij}, and the sense preference of word j, (x_j). During each phase of the dynamics, a process of selection allows strategies with higher payoff to emerge and at the end of the process each player chooses its sense according to these constraints.

The complexity of each step of the replicator dynamics is quadratic but there are different dynamics that can be used with our framework to solve the problem more efficiently, such as the recently introduced *infection and immunization* dynamics (Rota Buló, Pelillo, and Bomze 2011), which have a linear-time/space complexity per step and are known to be much faster than, and as accurate as, the replicator dynamics.

5.2 Implementation Details

In this section we describe the association measures used to weight the graph W (Section 5.2.1), the semantic and relatedness measures used to compare the synsets (Section 5.2.2), the computation of the payoff matrices of the games (Section 5.2.3), and the different implementations of the system strategy space (Section 5.2.4) in cases of unsupervised, semi-supervised, and coarse-grained WSD.

5.2.1 Association Measures. We evaluated our algorithm with different similarity measures in order to find the measure that performs better; the results of this evaluation

$$dice = \frac{2O_{11}}{R_1+C_1} \qquad m\text{-}dice = log_2 O_{11} \frac{2O_{11}}{R_1+C_1} \qquad pmi = log_2 \frac{0}{E_{11}}$$

$$t\text{-}score = \frac{O-E_{11}}{\sqrt{O}} \qquad odds\text{-}r = log \frac{(O_{11}+1/2)(O_{22}+1/2)}{(O_{12}+1/2)(O_{21}+1/2)} \qquad z\text{-}score = \frac{O-E_{11}}{\sqrt{E_{11}}}$$

$$chi\text{-}s = \sum_{ij} \frac{(O_{ij}-E_{ij})^2}{E_{ij}} \qquad chi\text{-}s\text{-}c = \frac{N(|O_{11}O_{22}-O_{12}O_{21}|-N/2)^2}{R_1R_2C_1C_2}$$

Figure 2
Association measures used to weight the co-occurrence graph *W*.

	w_j	$\neg w_j$	
w_i	O_{11}	O_{12}	$= R_1$
$\neg w_i$	O_{21}	O_{22}	$= R_2$
	$= C_1$	$= C_2$	$= N$

	w_j	$\neg w_j$
w_i	$E_{11} = R_1C_1/N$	$E_{12} = R_1C_2/N$
$\neg w_i$	$E_{21} = R_2C_1/N$	$E_{22} = R_2C_2/N$

Figure 3
Contingency tables of observer frequency (on the left) and expected frequency (on the right).

are presented in Section 6.2.1. Specifically, for our experiments, we used eight different measures: the *Dice coefficient* (*dice*) (Dice 1945), the *modified Dice coefficient* (*mDice*) (Kitamura and Matsumoto 1996), the pointwise mutual information (*pmi*) (Church and Hanks 1990), the *t-score* measure (*t-score*) (Church and Hanks 1990), the *z-score* measure (*z-score*) (Burrows 2002), the *odds ration* (*odds-r*) (Blaheta and Johnson 2001), the *chi-squared* test (*chi-s*) (Rao 2002), and the *chi-squared* correct (*chi-s-c*) (DeGroot et al. 1986).

The measures that we used are presented in Figure 2, where the notation refers to the standard contingency tables (Evert 2008) used to display the observed and expected frequency distribution of the variables, respectively, on the left and on the right of Figure 3. All the measures for the experiments in this article have been calculated using the BNC corpus (Leech 1992) because it is a well balanced general domain corpus.

5.2.2 Semantic and Relatedness Measures. We used WordNet (Miller 1995) and BabelNet (Navigli and Ponzetto 2012a) as knowledge bases to collect the sense inventories of each word to be disambiguated.

Semantic and Relatedness Measures Calculated with WordNet. WordNet (Miller 1995) is a lexical database where the lexicon is organized according to a psycholinguistic theory of the human lexical memory, in which the vocabulary is organized conceptually rather than alphabetically, giving a prominence to word meanings rather than to lexical forms. The database is divided in five parts: nouns, verbs, adjectives, adverbs, and functional words. In each part the lexical forms are mapped to the senses related to them; in this way it is possible to cluster words that share a particular meaning (synonyms) and to create the basic component of the resource: the **synset**. Each synset is connected in a network to other synsets, which have a semantic relation with it.

The relations in WordNet are: hyponymy, hypernymy, antonymy, meronymy, and holonymy. **Hyponymy** gives the relations from more general concepts to more specific; **hypernymy** gives the relations from particular concepts to more general; **antonymy** relates two concepts that have an opposite meaning; **meronymy** connects the concept that is part of a given concept with it; and **holonymy** relates a concept with its constituents. Furthermore, each synset is associated with a definition and gives the morphological relations of the word forms related to it. Given the popularity of the resource many parallel projects have been developed. One of them is eXtended WordNet (Mihalcea and Moldovan 2001), which gives a parsed version of the glosses together with their logical form and the disambiguation of the term in it.

We have used this resource to compute similarity and relatedness measures in order to construct the payoff matrices of the games. The computation of the sense similarity measures is generally conducted using relations of likeness such as the *is-a* relation in a taxonomy; on the other hand, the relatedness measures are more general and take into account a wider range of relations such as the *is-a-part-of* or *is-the-opposite-of*.

The semantic similarity measures that we used are the *wup similarity* (Wu and Palmer 1994) and the *jcn measure* (Jiang and Conrath 1997). These measures are based on the structural organization of WordNet and compute the similarity among the two senses s_i, s_j according to the depth of the two senses in the lexical database and that of the most specific ancestor node (*msa*) of the two senses. The *wup similarity*, described in Equation (12), takes into account only the path length among two concepts. The *jcn measure* combines corpus statistics and structural properties of a knowledge base. It is computed as presented in Equation (13), where *IC* is the information content of a concept *c* derived from a corpus[3] and computed as $IC(c) = log^{-1}P(c)$.

$$ssim_{wup}(s_i, s_j) = 2 * depth(msa)/(depth(s_i) + depth(s_j)) \tag{12}$$

$$ssim_{jcn}(s_i, s_j) = IC(s_1) + IC(s_2) - 2IC(msa) \tag{13}$$

The semantic relatedness measures that we used are based on the computation of the similarity among the definitions of two concepts in a lexical database. These definitions are derived from the glosses of the synsets in WordNet. They are used to construct a co-occurrence vector $v_i = (w_{1,i}, w_{2,i} ... w_{n,i})$ for each concept *i*, with a bag-of-words approach where *w* represents the number of times word *w* occurs in the gloss and *n* is the total number of different words (*types*) in the corpus.[4] This representation allows us to project each vector into a *vector space*, where it is possible to conduct different kinds of computations. For our experiments, we decided to calculate the similarity among two glosses using the cosine distance among two vectors, as shown in Equation (14), where the nominator is the intersection of the words in the two glosses and $||v||$ is the norm of the vectors, which is calculated as: $\sqrt{\sum_{i=1}^{n} w_i^2}$.

$$\cos\theta \frac{v_i \cdot v_j}{||v_i||||v_j||} \tag{14}$$

[3] We used the IC files computed on SemCor (Miller et al. 1993) for the experiments in this article. They are available at `http://wn-similarity.sourceforge.net` and are mapped to the corresponding version of WordNet of each data set.

[4] In our case the corpus is composed of all the WordNet glosses.

This measure gives the cosine of the angle between the two vectors and, in our case, returns values ranging from 0 to 1 because the values in the co-occurrence vectors are all positive. Given the fact that small cosine distances indicate a high similarity, we transform this distance measure into a similarity measure with $1 - cos(v_i, v_j)$.

The procedure to compute the semantic relatedness of two synsets has been introduced by Patwardhan and Pedersen (2006) as *Gloss Vector measure*; and we used it with four different variations for our experiments. The four variations are named: tf-idf, tf-idf$_{\text{ext}}$, vec, and vec$_{\text{ext}}$. The difference among them relies on the way the gloss vectors are constructed. Because the synset gloss is usually short we used the concept of *super-gloss* as in Patwardhan and Pedersen (2006) to construct the vector of each synset. A **super-gloss** is the concatenation of the gloss of the synset plus the glosses of the synsets, which are connected to it via some WordNet relations (Pedersen 2012). We used the WordNet version that has been used to label each data set. Specifically, the different implementations of the vector construction vary on the way in which the co-occurrence is calculated, the corpus used, and the source of the relations. *tf-idf* constructs the co-occurrence vectors exploiting the *term frequency - inverse document frequency* weighting schema (*tf-idf*). *tf-idf$_{ext}$* uses the same information of *tf-idf* plus the relations derived from eXtended WordNet (Mihalcea and Moldovan 2001). *vec* uses a standard bag-of-words approach to compute the co-occurrences. *vec$_{ext}$* uses the same information of *vec* plus the relations from eXtended WordNet.

Instead of considering only the raw frequency of terms in documents, the *tf-idf* method scales the importance of less informative terms taking into account the number of documents in which a term occurs. Formally, it is the product of two statistics: the term frequency and the inverse document frequency. The former is computed as the number of times a term occurs in a document (gloss in our case); the latter is computed as $idf_t = \log \frac{N}{df_t}$, where N is the number of documents in the corpus and df_t is the number of documents in which the term occurs.

Relatedness Measure Calculated with BabelNet and NASARI. BabelNet (Navigli and Ponzetto 2012a) is a wide-coverage multilingual semantic network. It integrates lexicographic and encyclopedic knowledge from WordNet and Wikipedia, automatically mapping the concepts shared by the two knowledge bases. This mapping generates a semantic network where millions of concepts are lexicalized in different languages. Furthermore, it allows linking *named entities*, such as *Johann Sebastian Bach*, and concepts, such as *composer* and *organist*.

BabelNet can be represented as a labeled direct graph $G = (V, E)$ where V is the set of nodes (*concepts* or *named entities*) and $E \subseteq V \times R \times V$ is the set of edges connecting pairs of *concepts* or *named entities*. The edges are labeled with a semantic relation from R, such as: *is-a*, *given name*, or *occupation*. Each node $v \in V$ contains a set of lexicalizations of the concept for different languages, which forms a BabelNet synset.

The semantic measure, which we developed using BabelNet, is based on NASARI[5] (Camacho-Collados, Pilehvar, and Navigli 2015), a semantic representation of the *concepts* and *named entities* in BabelNet. This approach first exploits the BabelNet network to find the set of related *concepts* in WordNet and Wikipedia and then constructs two vectors to obtain a semantic representation of a concept b. These representations are projected in two different semantic spaces, one based on words and the other on synsets.

[5] The resource is available at `http://lcl.uniroma1.it/nasari/`.

They use lexical specificity[6] (Lafon 1980) to extract the most representative words to use in the first vector and the most representative synsets to use in the second vector.

In this article, we computed the similarity between two senses using the vectors (of the word-based semantic space) provided by NASARI. These semantic representations provide for each sense the set of words that best represent the particular concept and the score of representativeness of each word. From this representation we computed the pairwise cosine similarity between each concept, as described in the previous section for the semantic relatedness measures.

The use of NASARI is particularly useful in the case of named entity disambiguation because it includes many entities that are not included in WordNet. On the other hand, it is difficult to use it in all-words sense disambiguation tasks, since it includes only WordNet synsets that are mapped to Wikipedia pages in BabelNet. For this reason it is not possible to find the semantic representation for many verbs, adjectives, and adverbs that are commonly found in all-words sense disambiguation tasks.

We used the SPARQL endpoint[7] provided by BabelNet to collect the sense inventories of each word in the texts of each data set. For this task we filtered the first 100 resources whose label contains the lexicalization of the word to be disambiguated. This operation is required because in many cases it is possible to have indirect references to entities.

5.2.3 From Similarities to Payoffs. The similarity and relatedness measures are computed for all the senses of the words to be disambiguated. From this computation it is possible to obtain a similarity matrix Z that incorporates the pairwise similarity among all the possible senses. This computation could have heavy computational cost, if there are many words to be disambiguated. To overcome this issue, the pairwise similarities can be computed just one time on the entire knowledge base and used in actual situations, reducing the computational cost of the algorithm. From this matrix we can obtain the partial semantic similarity matrix for each pair of players, $Z_{ij} = m \times n$, where m and n are the senses of i and j in Z.

5.2.4 Strategy Space Implementation. Once the pairwise similarities between the words and their senses, stored in the two matrices W and Z, are calculated, we can pass to the description of the strategy space of each player. It can be initialized with Equation (15), which follows the constraints described in Section 4.2 and assigns to each sense an equal probability.

$$s_{ij} = \begin{cases} |M_i|^{-1}, & \text{if sense } j \text{ is in } M_i \\ 0, & \text{otherwise} \end{cases} \tag{15}$$

This initialization is used in the case of unsupervised WSD because it does not use any prior knowledge about the senses distribution. In case we want to exploit information from prior knowledge, obtained from sense-labeled data, we can assign to each sense a probability according to its rank, concentrating a higher probability on senses that have a high frequency. To model this kind of scenario we used a geometric distribution

[6] A statistical measure based on the hypergeometric distribution over word frequencies.

[7] `http://babelnet.org/sparql/`.

that gives us a decreasing probability distribution. This new initialization is defined as follows:

$$s_{ij} = \begin{cases} p(1-p)^{r_j}, & \text{if sense } j \text{ is in } M_i \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

where p is the parameter of the geometric distribution and determines the scale or statistical dispersion of the probability distribution, and r_j is the rank of sense j, which ranges from 1 (the rank of the most common sense) to m (the rank of the least frequent sense). Finally, the vector obtained from Equation (16) is divided by $\sum_{j \in S_i} p_j$ in order to make the probabilities add up to 1. In our experiments, we used the ranked system provided by the Natural Language Toolkit (version 3.0) (Bird 2006) to rank the senses associated with each word to be disambiguated. Natural Language Toolkit is a suite of modules and data sets covering symbolic and statistical NLP. It includes a WordNet reader that can be queried with a lemma and a part of speech to obtain the list of possible synsets associated with the specified lemma and the part of speech. The returned synsets are listed in decreasing order of frequency and can be used as ranking systems by our algorithm.

We used the method proposed by Navigli (2006) for the experiments on coarse-grained WSD. With this approach it is possible to cluster the senses of a given word according to the similarity that the senses share. In this way it is possible to obtain a set of disjoint clusters $O = \{o_1, ..., o_t\}$, which is ranked according to the information obtained with the ranking system described earlier for each sense inventory M. The initialization of the strategy space, in this case, is defined as follows:

$$s_{ij} = \begin{cases} p(1-p)^{r_o}, & \text{if sense } j \text{ is in cluster } o \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

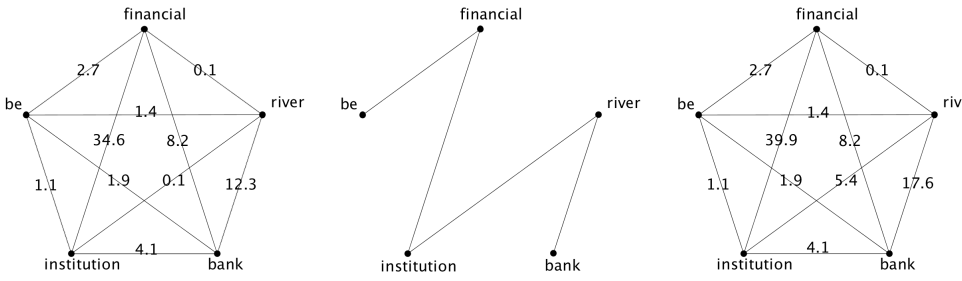
With this initialization it is possible to assign an equal probability to the senses belonging to a determined cluster and to rank the clusters according to the ranking of the senses in each of them.

5.3 An Example

As an example, we can consider the following sentence, which we encountered before:

- There is a financial institution near the river bank.

We first tokenize, lemmatize, and tag the sentence; then we extract the content words that have an entry in WordNet 3.0 (Miller 1995), constructing the list of words to be disambiguated: {is, financial, institution, river, bank}. Once we have identified the target words we compute the pairwise similarity for each target word. For this task we use the Google Web 1T 5-Gram Database (Brants and Franz 2006) to compute the



(a) Co-occurrence graph (b) *n*-gram graph (c) Similarity *n*-gram graph

Figure 4
Three graph representations for the sentence: *there is a financial institution near the river bank*. (a) A co-occurrence graph constructed using the modified Dice coefficient as similarity measure over the Google Web 1T 5-Gram Database (Brants and Franz 2006) to weight the edges. (b) Graph representation of the *n*-gram structure of the sentence, with $n = 1$; for each node, an edge is added to another node if the corresponding word appears to its left or right in a window the size of one word. (c) A weighted graph that combines the information of the co-occurrence graph and the *n*-gram graph. The edges of the co-occurrence graph are augmented by its mean weight if a corresponding edge exists in the *n*-gram graph and does not include a stop-word.

modified Dice coefficient[8] (Kitamura and Matsumoto 1996). With the information derived by this process we can construct a co-occurrence graph (Figure 4(a)), which indicates the strength of association between the words in the text. This information can be augmented, taking into account other sources of information such that the dependency structure of the syntactic relations between the words[9] or the proximity information derived by a simple *n*-gram model (Figure 4(b), $n = 1$).

The operation to increment the weights of structurally related words is important because it prevents the system from relying only on distributional information, which could lead to a sense shift for the ambiguous word *bank*. In fact, its association with the words *financial* and *institution* would have the effect of interpreting it as a *financial institution* and not as *sloping land*, as defined in WordNet. Furthermore, using only distributional information could exclude associations between words that do not appear in the corpus in use.

In Figure 4(c) we see the final form of the graph for our target sentence, in which we have combined the information from the co-occurrence graph and from the *n*-gram graph. The weights in the co-occurrence graph are increased by the mean weight of the graph if a corresponding edge exists in the *n*-gram graph and does not include a stop-word.[10]

[8] Specifically we used the service provided by the Corpus Linguistics group at FAU Erlangen-Nürnberg, with a collocation span of four words on the left and on the right and collocates with minimum frequency: 100.

[9] This aspect is not treated in this article.

[10] A more accurate representation of the data can be obtained using the dependency structure of the sentence instead of the *n*-gram graph; but in this case the results would not have changed, since in both cases there is an edge between *river* and *bank*. In fact, in many cases a simple *n*-gram model can implicitly detect syntactical relations. We used the stop-word list available in the Python Natural Language Toolkit, described earlier.

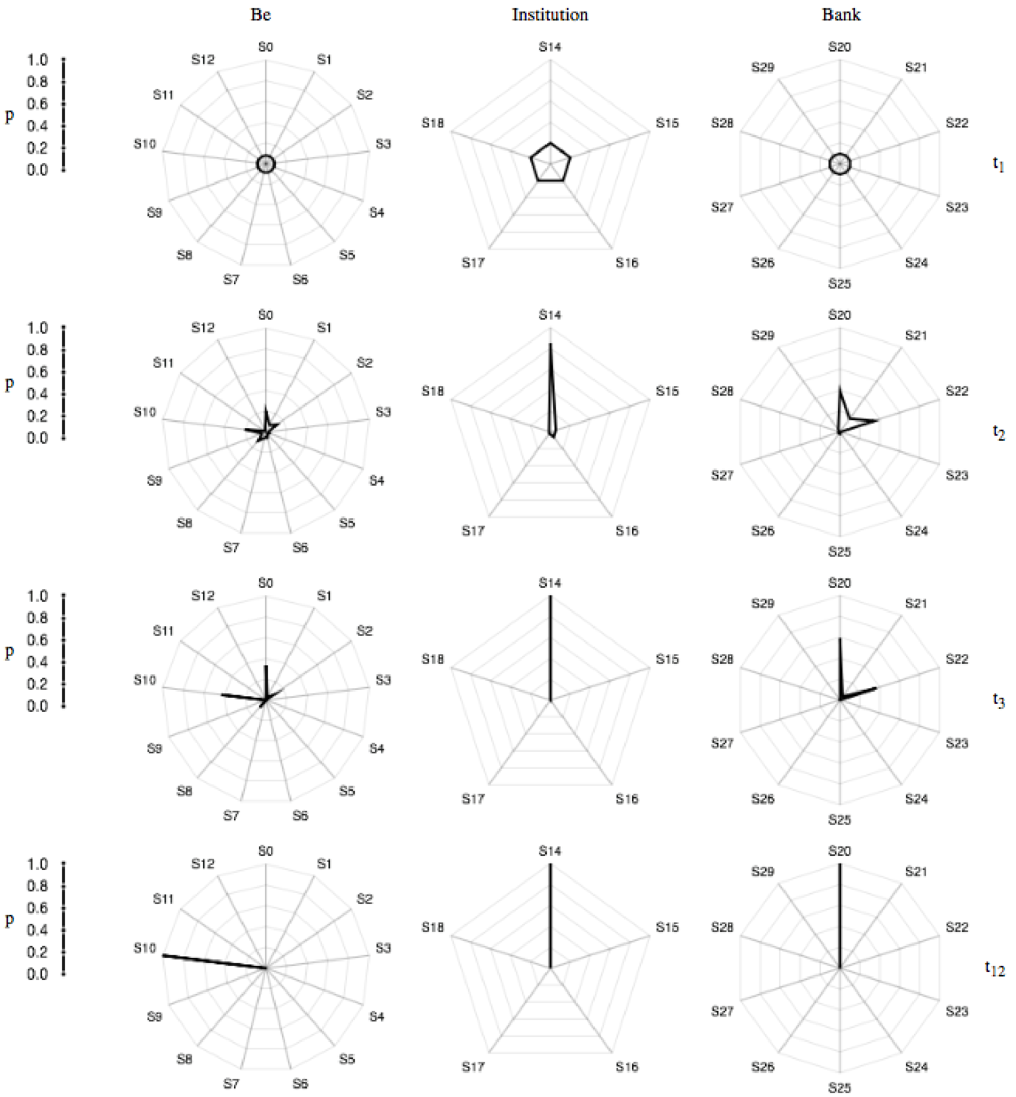


Figure 5
System dynamics for the words: *be*, *institution*, and *bank* at time step 1, 2, 3, and 12 (system convergence). The strategy space of each word is represented as a regular polygon of radius 1, where the distance from the center to any vertex represents the probability associated with a particular word sense. The values on each radius in a polygon are connected with a darker line in order to show the actual probability distribution obtained at each time step.

After the pairwise similarities between the words are computed, we access a lexical database in order to obtain the sense inventories of each word so that each word can be associated with a predefined number of senses. For this task, we use WordNet 3.0 (Miller 1995). Then, for each unique sense in all the sense inventories, we compute the pairwise semantic similarity in order to identify the affinity among all the pairwise sense combinations. This task can be done using a semantic similarity or relatedness measure.[11] For this example, we used a variant of the *gloss vector measure* (Patwardhan and Pedersen 2006), the *tf-idf*, described in Section 5.2.2.

Having obtained the similarity information, we can initialize the strategy space of each player with a uniform distribution, given the fact that we are not considering any prior information about the senses distributions. Now the system dynamics can be started. In each iteration of the dynamics each player plays a game with its neighbors, obtaining a payoff for each of its strategies according to Equation (10); once the players have played the games with their neighbors in W, the strategy space of each player is updated at time $t+1$ according to Equation (6).

We present the dynamics of the system created for the example sentence in Figure 5. The dynamics are shown only for the ambiguous words at time steps t_1, t_2, t_3, and t_{12} (when the system converges). As we can see, at time step 1 the senses of each word are equiprobable, but as soon as the games are played some senses start to emerge. In fact at time step 2 many senses are discarded, and this in virtue of two principles: a) the words in the text push the senses of the other words toward a specific sense; and b) the sense similarity values for certain senses are very low. Regarding the first principle, we can consider that the word *institution*, which is playing the games with the words *financial* and *bank*, is immediately driven toward a specific sense, as an organization founded and united for a specific purpose as defined in WordNet 3.0—thus discarding the other senses. Regarding the second principle, we can consider many senses of the word *bank* that are not compatible with the senses of the other words in the text, and therefore their values decrease rapidly.

The most interesting phenomenon that can be appreciated from the example is the behavior of the strategy space of the word *bank*. It has ten senses according to WordNet 3.0 (Miller 1995), and can be used in different contexts and domains to indicate, among other things, a financial institution (s_{22} in Figure 5) or a sloping land (s_{20} in Figure 5). When it plays a game with the words *financial* and *institution*, it is directed toward its financial sense; when it plays a game with the word *river*, it is directed toward its naturalistic meaning. As we can see in Figure 5 at time step 2, the two meanings (s_{20} and s_{22}) have almost the same value and at time step 3 the word starts to define a precise meaning to the detriment of s_{21} but not of s_{22}. The balancing of these forces toward a specific meaning is given by the similarity value w_{ij}, which allows *bank* in this case to choose its naturalistic meaning. Furthermore, we can see that the inclination to a particular sense is given by the payoff matrix Z_{ij} and by the strategy distribution S_j, which indicates what sense word j is going to choose, ensuring that word i's is coherent with this choice.

6. Experimental Evaluation

We now describe how the parameters of the presented method have been found and how it has been tested and compared with state-of-the-art systems[12] in Section 6.1 and Section 6.2, respectively. We describe the data sets used for the tuning and for the evaluation of our model and the different settings used to test it. The results of our experiments using WordNet as knowledge base are described in Section 6.2.1, where

[11] Semantic similarity and relatedness measures are discussed in Sections 5.2.1 and 5.2.2.

[12] The code of the algorithm and the data sets used are available at `http://www.dsi.unive.it/∼tripodi/wsd`.

two different implementations of the system are proposed—the unsupervised and the supervised. In Section 6.2.1 we compare our results with state-of-the-art systems. Finally, the results of the experiments using BabelNet as knowledge base, related to WSD and entity disambiguation, are described in Section 6.2.2. The results are provided as F1, computed according to the following equation:

$$\text{F1} = 2 \cdot \frac{\textit{precision} \cdot \textit{recall}}{\textit{precision} + \textit{recall}} \times 100 \tag{18}$$

F1 is a measure that determines the weighted harmonic mean of precision and recall. Precision is defined as the number of correct answers divided by the number of provided answers; and recall is defined as the number of correct answers divided by the total number of answers to be provided.

6.1 Parameter Tuning

We used two data sets to tune the parameters of our approach, SemEval-2010 task 17 (S10) (Agirre et al. 2009) and SemEval-2015 Task 13 (S15) (Moro and Navigli 2015). The first data set is composed of three English texts from the ecology domain for a total of 1,398 words to be disambiguated (1,032 nouns/named entities and 366 verbs). The second data set is composed of four English documents, from different domains: medical, drug, math, and social issues, for a total of 1,261 instances, including nouns/named entities, verbs, adjectives, and adverbs. Both data sets have been manually labeled using WordNet 3.0. The only difference between these data sets is that the target words of the first data set belong to a specific domain, whereas all the content words of the second data set have to be disambiguated. We used these two typologies of data set to evaluate our algorithm in different scenarios; furthermore, we created, from each data set, 50 different data sets, selecting from each text a random number of sentences and evaluating our approach on each of these data sets to identify the parameters that on average perform better than others. In this way it is possible to simulate a situation in which the system has to work on texts of different sizes and on different domains. This because, as demonstrated by Søgaard et al. (2014), the results of a determined algorithm are very sensitive to sample size. The number of target words for each text in the random data sets ranges from 12 to 571. The parameters that will be tuned are the association and semantic measures to use to weight the similarity among words and senses (Section 6.1.1), the n of the n-gram graph used to increase the weights of near words (Section 6.1.2), and the p of the geometric distribution used by our semi-supervised system (Section 6.1.3).

6.1.1 Association and Semantic Measures. The first experiment that we present is aimed at finding the semantic and distributional measures with the highest performances. Recall that we used WordNet 3.0 as knowledge base and the BNC corpus (Leech 1992) to compute the association measures. In Tables 2 and 3 we report the average results on the S10 and S15 data sets, respectively. From these tables it is possible to see that the performance of the system is highly influenced by the combination of measures used. As an example of the different representations generated by the measures described

Table 2
Results as F1 for S10. The first result with a statistically significant difference from the best (**bold** result) is marked with * (χ^2, $p < 0.05$).

	dice	mdice	pmi	t-score	z-score	odds-r	chi-s	chi-s-c
tfidf	55.5	56.3	50.6	45.4	50.1	49.8	39.1	54.4
tfidf$_{ext}$	**56.5**	55.9	50.1	45.0	49.9	49.5	39.1	54.2
vec	54.7	54.3	49.3	44.1	49.4	53.6	39.3	50.5
vec$_{ext}$	55.0	54.3	48.8	43.8	48.6	53.6	39.1	49.9
jcn	51.3	50.6	40.1	50.1	47.6	52.6*	50.1	50.6
wup	37.2	36.9	35.6	32.2	37.9	36.8	38.4	35.4

Table 3
Results as F1 for S15. The first result with a statistically significant difference from the best (**bold** result) is marked with * (χ^2, $p < 0.05$).

	dice	mdice	pmi	t-score	z-score	odds-r	chi-s	chi-s-c
tfidf	64.1	64.2	63.1	59.0	61.8	**65.3**	63.3*	62.4
tfidf$_{ext}$	62.9	63.1	62.4	58.7	60.9	63.0	62.0	61.1
vec	62.8	62.3	62.8	59.8	62.3	62.9	61.1	60.3
vec$_{ext}$	60.5	59.9	61.2	57.8	59.7	60.6	60.1	59.4
jcn	57.2	57.6	56.7	57.9	57.0	56.9	57.5	57.6
wup	46.2	45.4	43.8	45.4	45.9	47.4	46.1	45.5

in Section 5.2, we can observe Figures 6 and 7, which depict the matrices Z and the adjacency matrix of the graph W, respectively, and are computed on the following three sentences from the second text of S10:

> The rivers Trent and Ouse, which provide the main fresh water flow into the Humber, drain large industrial and urban areas to the south and west (River Trent), and less densely populated agricultural areas to the north and west (River Ouse). The Trent/Ouse confluence is known as Trent Falls. On the north bank of the Humber estuary the principal river is the river Hull, which flows through the city of Kingston-upon-Hull and has a tidal length of 32 km up to the Hempholme Weir.

resulting in the following 35 content words (names and verbs) and 131 senses.

1. river n
2. Trent n
3. Ouse n
4. provide v
5. main n
6. water n
7. flow n
8. Humber n
9. drain v
10. area n
11. south n
12. west n
13. River n
14. Trent n
15. area n
16. River n
17. Ouse n
18. Trent n
19. Ouse n
20. confluence n
21. be v
22. Trent n
23. Falls n
24. bank n
25. Humber n
26. estuary n
27. river n
28. be v
29. river n
30. flow v
31. city n
32. have v
33. length* n
34. km n
35. Weir n

The first observation that can be made on the results is related to the semantic measures; in fact, the relatedness measures perform significantly better than the semantic similarity measures. This is because *wup* and *jcn* can be computed only on synsets

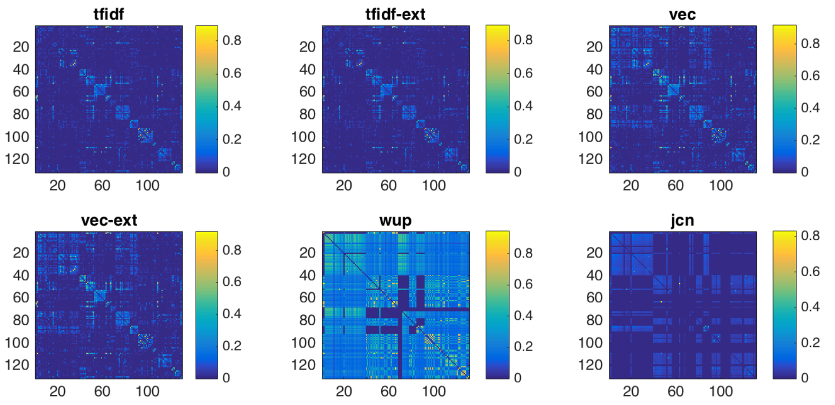


Figure 6
The representations of the payoff matrix Z computed on three sentences of the second text of S10, with the measures described in Section 5.2.2. All the senses of the words in the text are sequentially ordered.

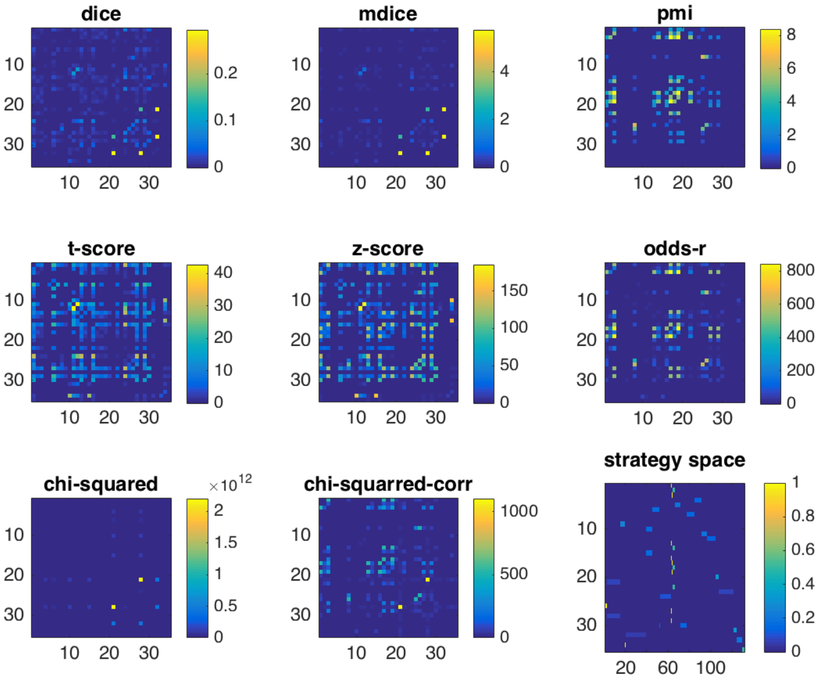
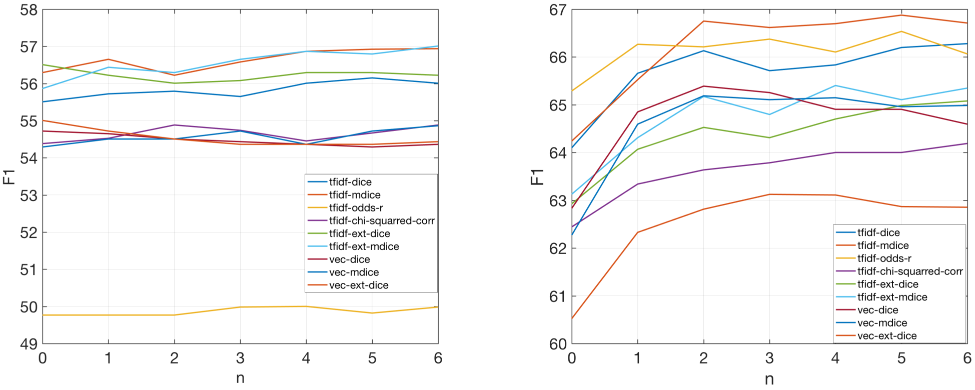


Figure 7
The representations of the adjacency matrix of the graph W computed on three sentences of the second text of S10, with the measures described in Section 5.2.1. The words are ordered sequentially and reflect the list proposed in the text. For a better visual comparison only positive values are presented, whereas the experiments are performed considering also negative values. The last image represents the strategy space of the players.

that have the same part of speech. This limitation affects the results of the algorithm because the games played between two words with different parts of speech have no effect on the dynamics of the system, since the values of the resulting payoff matrices are all zeros. This affects the performances of the system in terms of recall—because in this situation these words tend to remain on the central point of the simplex—and also in terms of precision—because the choice of the meaning of a word is computed only taking into account the influence of words with the same part of speech. In fact, from Figure 6 we can see that the representations provided by *wup* and *jcn* for the text described above have many uniform areas; this means that these approaches are not able to provide a clear representation of the data. On the contrary, the representations provided by the relatedness measures show a block structure on the main diagonal of the matrix, which is exactly what is required for a similarity measure. The use of the *tf-idf* weighting schema seems to be able to reduce the noise in the data representation; in fact the weights on the left part of the matrix are reduced by *tfidf* and *tfidf-ext* whereas they have high values in *vec* and *vec-ext*. The representations obtained with eXtended WordNet are very similar to those obtained with WordNet and their performance is also very close, although on average WordNet outperforms eXtended WordNet.

If we observe the performances of the association measures we notice that on average the best measures are *dice*, *mdice*, *chi-s-c*, and also *odds-r* on S15—the other measures perform almost always under the statistical significance. Observing the representations in Figure 7 we can see that *dice* and *mdice* have a similar structure; the difference between these two measures are that *mdice* has values on a different range and tends to better differentiate the weights, whereas in *dice* the values are almost uniform. *Pmi* tends to take high values when one word in the collocation has low frequency, but this does not imply high dependency, thus it compromises the results of the games. From its representation we can observe that its structure is different from the previous two—in fact, it concentrate its values on collocations such as *river Trent* and *river Ouse* and this has the effect of unbalancing the data representation. In fact, the *dice* and *mdice* concentrate their values on collocations such as *river flow* and *bank estuary*. *T-score* and *z-score* have a similar structure, the only difference is in the range of the values. For these measures we can see that the distribution of the values is quite homogeneous, meaning that these measures are not able to balance the weights well. On *odds-r* we recognize a structure similar to that of *pmi*, the main difference being that it works on a different range. The values obtained with *chi-s* are on a wide range, which compromises the data representation; in fact, its results are always under the statistical significance. *Chi-s-c* works on a narrower range than *chi-s* and its structure resembles that of *dice*—in fact, its results are often high.

6.1.2 n*-gram Graph.* The association measures are able to provide a good representation of the text but in many cases it is possible that a word in a specific text is not present in the corpus on which these measures are calculated; furthermore, it is possible that these words are used with different lexicalizations. One way to overcome these problems is to increase the values of the nodes near a determined word; in this way it is possible to ensure that the nodes in W are always connected. Furthermore, this allows us to exploit local information, increasing the importance of the words that share a proximity relation with a determined word; in this way it is possible to give more importance to (possibly syntactically) related words, as described in Section 5.1.1. To test the influence that the parameter of the n-gram graph has on the performance of the algorithm we selected the association and relatedness measures with the highest results and conducted a series of experiments on the same data sets presented above, with increasing values of n. The results of these experiments on S10 and S15 are presented in Figure 8(a) and 8(b), respectively. From the plots we can see that this approach is always beneficial for S15 and that the results increased substantially with values of n greater than 2. To the contrary, on S10 this approach is not always beneficial but in many cases it is



(a) S10

(b) S15

Figure 8
Results as F1 on S10 (on the left) and S15 (on the right) with increasing values of neighbor nodes (n).

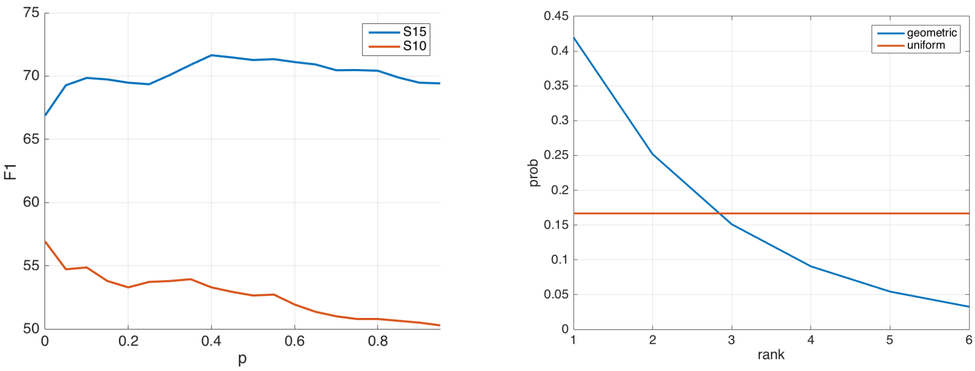
possible to notice an improvement. In particular, we notice that the pair of measures with highest results on both data sets is *tfidf-mdice* with $n = 5$. This also confirms our earlier experiments in which we saw that these two measures are particularly suited for our algorithm.

6.1.3 Geometric Distribution. Once we have identified the measures to use in our unsupervised system, we can test for the best parameter to use in case we want to exploit information from sense-labeled corpora. To tune the parameter of the geometric distribution (described in Section 5.2.4), we used the pair of measures and the value of n detected with the previous experiments and ran the algorithm on S10 and S15 with increasing values of p, in the interval $[0.05, 0.95]$.

The results of this experiment are presented in Figure 9(a), where we can see that the performance of the semi-supervised system on S15 is always better than that obtained with the unsupervised system ($p = 0$). On the other hand, the performance on S10 is always lower than that obtained with the unsupervised system. This behavior is not surprising because the target words of S10 belong to a specific semantic domain. We used SemCor to obtain the information about the sense distributions and this resource is a general domain corpus, which is not tailored for this specific task. In fact, as pointed out by McCarthy et. al (2007), the distribution of word senses on specific domains is highly skewed; for this reason, the most frequent sense heuristic calculated on general domains corpora, such as SemCor, is not beneficial for this kind of text.

From the plot we can see that on S15 the highest results are obtained with values of p ranging from 0.4 to 0.7 and for the evaluation of our model we decided to use $p = 0.4$ as parameter for the geometric distribution, because with this value we obtained the highest result.

6.1.4 Error Analysis. The main problems that we noticed analyzing the results of previous experiments are related to the semantic measures. As we pointed out in Section 6.1.1, these measures can be computed only on synsets with the same part of speech and this influences the results in terms of recall. The adverbs and adjectives are not disambiguated with these measures because of the lack of payoffs. This does not happen only



(a) Tuning. (b) Distributions.

Figure 9
Results as F1 on S10 and S15 with increasing values of p (on the left), $p = 0$ corresponds to the results with the unsupervised setting (on the left). An example of geometric distribution with six ranked senses compared with the uniform distribution (on the right).

in the case of function words with low semantic content but also for verbs with a rich semantic content, such as *generate*, *prevent*, and *obtain*. The use of the relatedness measures substantially reduces the number of words that are not disambiguated. With these measures, a word is not disambiguated only in cases in which the concepts denoted by it are not covered enough by the reference corpus—for example, in our experiments we have words such as *drawn-out*, *dribble*, and *catchment* that are not disambiguated.

To overcome this problem we have used the n-gram graph to increase the weights among neighboring words. Experimentally, we noticed that when this approach is used with the relatedness measures, it leads to the disambiguation of all the target words and with $n \geq 1$ we have $precision = recall$. The use of this approach influences the results also in terms of precision—in fact, if we consider the performance of the system on the word *actor*, we pass from F1 = 0 ($n = 0$) to F1 = 71.4 ($n = 5$). This is because the number of relations of the two senses (synsets) of the word *actor* are not balanced in WordNet 3.0; in fact, *actor* as *theatrical performer* has 21 relations whereas *actor* as *person who acts and gets things done* has only 8 relations, and this can compromise the computation of the semantic relatedness measures. It is possible to overcome this limitation using the local information given by the n-gram graph, which allows us to balance the influence of words in the text.

Another aspect to consider is whether the polysemy of the words influences the results of the system. Analyzing the results we noticed that the majority of the errors are made on words such as *make-v*, *give-v*, *play-v*, *better-a*, *work-v*, *follow-v*, *see-v*, and *come-v*, which have more that 20 different senses and are very frequent words difficult to disambiguate in fine-grained tasks. As we can see from Figure 10, this problem can be partially solved using the semi-supervised system. In fact, the use of information from sense-labeled corpora is particularly useful when the polysemy of the words is particularly high.

6.2 Evaluation Set-up

We evaluated our algorithm with three fine-grained data sets: Senseval-2 English all-words (S2) (Palmer et al. 2001), Senseval-3 English all-words (S3) (Snyder and Palmer

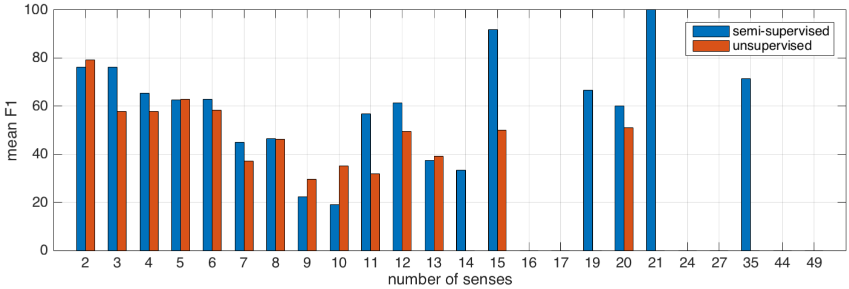


Figure 10
Average F1 on the words of S15 grouped by number of senses, using the unsupervised and the semi-supervised system.

2004), SemEval-2007 all-words (S7) (Pradhan et al. 2007), and one coarse-grained data set, SemEval-2007 English all-words (S7CG) (Navigli, Litkowski, and Hargraves 2007),[13] using as the knowledge base WordNet. Furthermore, we evaluated our approach on two data sets, SemEval-2013 task 12 (S13) (Navigli, Jurgens, and Vannella 2013) and KORE50 (Hoffart et al. 2012),[14] using as the knowledge base BabeNet.

We describe the evaluation using *WordNet* as the knowledge base in the next sections, and in Section 6.2.2 we present the evaluation conducted using *BabelNet* as the knowledge base. Recall that for all the next experiments we used *mdice* to weight the graph W, *tfidf* to compute the payoffs, $n = 5$ for the n-gram graph, and $p = 0.4$ in the case of semi-supervised learning. The results are provided as F1 for all the data sets except KORE50; for this data set the results are provided as accuracy, as is common in the literature.

6.2.1 Experiments Using WordNet as Knowledge Base. Table 4 shows the results as F1 for the four data sets that we used for the experiments with WordNet. The table includes the results for the two implementations of our system: the unsupervised and the semi-supervised and the results obtained using the most frequent sense heuristic. For the computation of the most frequent sense, we assigned to each word to be disambiguated the first sense returned by the WordNet reader provided by the Natural Language Toolkit (version 3.0) (Bird 2006). As we can see, the best performance of our system is obtained on nouns on all the data sets. This is in line with state-of-the-art systems because in general the nouns have lower polysemy and higher inter-annotator agreement (Palmer et al. 2001). Furthermore, our method is particularly suited for nouns. In fact, the disambiguation of nouns benefits from a wide context and local collocations (Agirre and Edmonds 2007).

We obtained low results on verbs on all data sets. This, as pointed out by Dang (1975), is a common problem not only for supervised and unsupervised WSD systems

[13] We downloaded S2 from `www.hipposmond.com/senseval2`, S3 from `http://www.senseval.org/senseval3`, S7 from `http://nlp.cs.swarthmore.edu/semeval/tasks/index.php`, and S7CG from `http://lcl.uniroma1.it/coarse-grained-aw`.

[14] We downloaded S13 from `https://www.cs.york.ac.uk/semeval-2013/task12/index.html` and KORE50 from `http://www.mpi-inf.mpg.de/departments/databases-and-information-systems/research/yago-naga/aida/downloads/`.

Table 4
Detailed results as F1 for the four data sets studied with *tf-idf* and *mdice* as measures. The results show the performance of our unsupervised (*uns*) and semi-supervised (*ssup*) system and the results obtained using the most frequent sense heuristic (MFS). Detailed information about the performance of the systems on different parts of speech are provided: nouns (N), verbs (V), adjectives (A), and adverbs (R).

SemEval 2007 coarse-grained - S7CG

Method	All	N	V	A	R
WSD^{uns}_{games}	80.4	85.5	71.2	81.5	76.0
$\text{WSD}^{ssup}_{games}$	82.8	85.4	77.2	82.9	84.6
MFS	76.3	76.0	70.1	82.0	86.0

SemEval 2007 fine-grained - S7

Method	All	N	V	A	R
WSD^{uns}_{games}	43.3	49.7	39.9	–	–
$\text{WSD}^{ssup}_{games}$	56.5	62.9	53.0	–	–
MFS	54.7	60.4	51.7	–	–

Senseval 3 fine-grained - S3

Method	All	N	V	A	R
WSD^{uns}_{games}	59.1	63.3	50.7	64.5	71.4
$\text{WSD}^{ssup}_{games}$	64.7	70.3	54.1	70.7	85.7
MFS	62.8	69.3	51.4	68.2	100.0

Senseval 2 fine-grained - S2

Method	All	N	V	A	R
WSD^{uns}_{games}	61.2	69.8	41.7	61.9	65.1
$\text{WSD}^{ssup}_{games}$	66.0	72.4	43.5	71.8	75.7
MFS	65.6	72.1	42.4	71.6	76.1

but also for humans, who in many cases disagree about what constitutes a different sense for a polysemous verb, compromising the sense tagging procedure.

As we anticipated in Section 6.1.3, the use of prior knowledge is beneficial for this kind of data set. As we can see in Table 4, using a semi-supervised setting improves the results of 5% on S2 and S3 and of 12% on S7. The large improvement obtained on S7 can be explained by the fact that the results of the unsupervised system are well below the most frequent sense heuristic, so exploiting the evidence from the sense-labeled data set is beneficial. For the same reason, the results obtained on S7CG with a semi-supervised setting are less impressive than those obtained with the unsupervised systems; in fact, the structure of the data sets is different and the results obtained with the unsupervised setting are well above the most frequent sense. This series of experiments confirms that the use of prior knowledge is beneficial in general domain data sets and that when it is used, the system performs better than the most common-sense heuristic computed on the same corpus.

Comparison to State-of-the-Art Algorithms. Table 5 shows the results of our system and the results obtained by state-of-the-art systems on the same data sets. We compared

Table 5
Comparison with state-of-the-art algorithms: unsupervised (*unsup.*), semisupervised (*semi sup.*), and supervised (*sup.*). *MFS* refers to the MFS heuristic computed on SemCor on each data set and *Best* refers to the best supervised system for each competition. The results are provided as F1 and the first result with a statistically significant difference from the best of each data set is marked with * (χ^2, $p < 0.05$).

		S7CG	S7CG (N)	S7	S3	S2
unsup.	*Nav10*	–	–	43.1	52.9	–
	PPR$_{w2w}$	80.1	83.6	41.7	57.9	59.7
	WSD$_{games}$	**80.4***	**85.5**	**43.3**	**59.1**	**61.2**
semi sup.	*IRST-DDD-00*	–	–	–	58.3	–
	MFS	76.3	77.4	54.7	62.8	65.6*
	MRF-LP	–	–	50.6*	58.6	60.5
	Nav05	**83.2**	84.1	–	60.4	–
	PPR$_{w2w}$	81.4	82.1	48.6	63.0	62.6
	WSD$_{games}$	82.8	**85.4**	**56.5**	**64.7***	**66.0**
sup.	*Best*	82.5	82.3*	**59.1**	65.2	**68.6**
	Zhong10	**82.6**	–	58.3	**67.6**	68.2

our method with supervised, unsupervised, and semi-supervised systems on four data sets. The supervised systems are *It Makes Sense* (Zhong and Ng 2010) (*Zhong10*); an open source WSD system based on support vector machines (Steinwart and Christmann 2008); and the best system that participated in each competition (*Best*). The semi-supervised systems are *IRST-DDD-00* (Strapparava, Gliozzo, and Giuliano 2004), based on WordNet domains and on manually annotated domain corpora; *MFS*, which corresponds to the most frequent sense heuristic implemented using the WordNet corpus reader of the natural language toolkit; *MRF-LP*, based on Markov random field (Chaplot, Bhattacharyya, and Paranjape 2015); *Nav05* (Navigli and Velardi 2005), a knowledge-based method that exploits manually disambiguated word senses to enrich the knowledge base relations; and *PPR*$_{w2w}$ (Agirre, de Lacalle, and Soroa 2014), a random walk method that uses contextual information and prior knowledge about senses distribution to compute the most important sense in a network given a specific word and its context. The unsupervised systems are: *Nav10*, a graph based WSD algorithm that exploits connectivity measures to determine the most important node in the graph composed by all the senses of the words in a sentence; and a version of the *PPR*$_{w2w}$ algorithm that does not use sense tagged resources.

The results show that our unsupervised system performs better than any other unsupervised algorithm in all data sets. In S7CG and S7, the difference is minimal compared with *PPR*$_{w2w}$ and *Nav10*, respectively; in *S3* and *S2*, the difference is more substantial compared with both unsupervised systems. Furthermore, the performance of our system is more stable on the four data sets, showing a constant improvement on the state of the art.

The comparison with semi-supervised systems shows that our system always performs better than the most frequent sense heuristic when we use information from sense-labeled corpora. We note strange behavior on S7CG: when we use prior knowledge, the performance of our semi-supervised system is lower than our unsupervised system and state of the art. This is because on this data set the performance of our unsupervised system is better than the results that can be achieved by using labeled data to initialize the strategy space of the semi-supervised system. On the other three data sets we note a substantial improvement in the performances of our system, with stable results higher than state-of-the-art systems.

Finally, we note that the results of our semi-supervised system on the fine-grained data sets are close to the performance of state-of-the-art supervised systems, with values that are statistically relevant only on $S3$. We also note that the performance of our system on the nouns of the S7CG data set is higher than the results of the supervised systems.

6.2.2 Experiments with BabelNet. BabelNet is particularly useful when the number of named entities to disambiguate is high. In fact, it is not possible to perform this task using only WordNet, because its coverage on named entities is limited. For the experiments on this section we used BabelNet to collect the sense inventories of each word to be disambiguated, the *mdice* measure to weight the graph W, and NASARI to obtain the semantic representation of each sense. The similarity among the representation obtained with this resource is computed using the cosine similarity measure, described in Section 5.2.2. The only difference with the experiments presented in Section 6.2.1 is that we used BabelNet as knowledge base and NASARI as resource to collect the sense representations instead of WordNet.

S13 consists of 13 documents in different domains, available in five languages (we used only English). All the nouns in these texts are annotated using BabelNet, with a total of 1,931 words to be disambiguated (English data set). KORE50 consists of 50 short English sentences with a total number of 146 mentions manually annotated using YAGO2 (Hoffart et al. 2013). We used the mapping between YAGO2 and Wikipedia to obtain for each mention the corresponding BabelNet concept, since there exists a mapping between Wikipedia and BabelNet. This data set contains highly ambiguous mentions that are difficult to capture without the use of a large and well-organized knowledge base. In fact, the mentions are not explicit and require the use of common knowledge to identify their intended meaning.

We used the SPARQL endpoint[15] provided by BabelNet to collect the sense inventories of the words in the texts of each data set. For this task we filtered the first 100 resources whose label contains the lexicalization of the word to be disambiguated. This operation can increase the dimensionality of the strategy space, but it is required because, particularly in KORE50, there are many indirect references—such as Tiger to refer to Tiger Woods (the famous golf player) or Jones to refer to John Paul Jones (the Led Zeppelin bassist).

Comparison to State-of-the-Art Algorithms. The results of these experiments are shown in Table 6, where it is possible to see that the performance of our system is close to the results obtained with Babelfy on S13 and substantially higher on KORE50. This is because with our approach it is necessary to respect the textual coherence, which is required when a sentence contains a high level of ambiguity, such as those proposed by KORE50. On the contrary, $\mathrm{PP}R_{w2w}$ performs poorly on this data set. This is because, as attested to in Moro, Raganato, and Navigli (2014), it disambiguates the words independently, without imposing any consistency requirements.

The good performance of our approach is also due to the good semantic representations provided by NASARI—in fact, it is able to exploit a richer source of information,

[15] `http://babelnet.org/sparql/`.

Table 6
Comparison with state-of-the-art algorithms on WSD and entity linking. The results are provided as F1 for S13 and as accuracy for KORE50. The first result with a statistically significant difference from the best (**bold** result) is marked with * (χ^2, $p < 0.05$).

	S13	KORE50
WSD_{games}	**70.8**	**75.7**
Babelfy	69.2	71.5
SUDOKU	66.3	–
MFS	66.5*	–
PPR_{w2w}	60.8	–
KORE	–	63.9*
GETALP	58.3	–

Wikipedia, which provides a larger coverage and a wider source of information than WordNet alone.

The results on KORE50 are presented as accuracy, following the custom of previous work on this data set. As we anticipated, it contains decontextualized sentences, which require common knowledge to be disambiguated. This common knowledge is obtained exploiting the relations in BabelNet that connect related entities, but in many cases this is not enough because the references to entities are too general and in this case the system is not able to provide an answer. It is also difficult to exploit distributional information on this data set because the sentences are short and in many cases cryptic. For these reasons the recall on this data set is well below the precision: 55.5%. The system does not provide answers for the entities in sentences such as: *Jobs and Baez dated in the late 1970s, and she performed at his Stanford memorial*, but it is able to correctly disambiguate the same entities in sentences where there is more contextual information.

7. Conclusions

In this article we introduced a new method for WSD based on game theory. We have provided an extensive introduction to the WSD task and explained the motivations behind the choice to model the WSD problem as a constraint-satisfaction problem. We conducted an extensive series of experiments to identify the similarity measures that perform better in our framework. We have also evaluated our system with two different implementations and compared our results with state-of-the-art systems, on different WSD tasks.

Our method can be considered as a continuation of knowledge-based, graph-based, and similarity-based approaches. We used the methodologies of these three approaches combined in a game theoretic framework. This model is used to perform a consistent labeling of senses. In our model we try to maximize the textual coherence, imposing that the meaning of each word in a text must be related to the meaning of the other words in the text. To do this we exploited distributional and proximity information to weight the influence that each word has on the others. We also exploited semantic similarity information to weight the strengths of compatibility among two senses. This is of great importance because it imposes constraints on the labeling process, developing a contextual coherence on the assignment of senses. The application of a game theoretic framework guarantees that these assumptions are met. Furthermore, the use of the replicator dynamics equation allows us to always find the best labeling assignment.

Our system, in addition to having a solid mathematical and linguistic foundation, has been demonstrated to perform well compared with state-of-the-art systems and to be extremely flexible. In fact, it is possible to implement new similarity measures, graph constructions, and strategy space initializations to test it in different scenarios. It is also possible to use it as completely unsupervised or to use information from sense-labeled corpora.

The features that make our system competitive compared with state-of-the-art systems are that instead of finding the most important sense in a network to be associated with the meaning of a single word, our system disambiguates all the words at the same time, taking into account the influence that each word has on the others and imposing sense compatibility among each sense before assigning a meaning. We have demonstrated how our system can deal with sense shifts, where a centrality algorithm, which tries to find the most important sense in a network, can be deceived by the context. In our case, the weighting of the context ensures respecting the proximity structure of a sentence and disambiguating each word according to the context in which it appears. This is because the meaning of a word in a sentence does not depend on *all* the words in the sentence but only on those that share a proximity (or syntactical) relation and those that enjoy a high distributional similarity.

Acknowledgments

This work was supported by the Samsung Global Research Outreach Program. We are deeply grateful to Rodolfo Delmonte for his insights on the preliminary phase of this work and to Bernadette Sharp for her help during the final part of it. We would also like to thank Phil Edmonds for providing us the correct version of the Senseval 2 data set.

References

Agirre, E., O. L. De Lacalle, C. Fellbaum, A. Marchetti, A. Toral, and P. Vossen. 2009. SemEval-2010 task 17: All-words word sense disambiguation on a specific domain. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 123-128, Boulder, CO.

Agirre, E., O. Lopez de Lacalle, and A. Soroa. 2014. Random walks for knowledge-based word sense disambiguation. *Computational Linguistics*, 40(1):57–84.

Agirre, E., O. Lopez De Lacalle, A. Soroa, and I. Fakultatea. 2009. Knowledge-based WSD and specific domains: Performing better than generic supervised WSD. In *Proceedings of IJCAI*, pages 1501–1506.

Agirre, E. and P. G. Edmonds. 2007. *Word Sense Disambiguation: Algorithms and Applications*, volume 33. Springer Science & Business Media.

Agirre, E., D. Martínez, O. L. de Lacalle, and A. Soroa. 2006. Two graph-based algorithms for state-of-the-art WSD. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 585–593, Sydney.

Araujo, L. 2007. How evolutionary algorithms are applied to statistical natural language processing. *Artificial Intelligence Review*, 28(4):275–303.

Bird, S. 2006. NLTK: The natural language toolkit. In *Proceedings of the COLING/ACL on Interactive Presentation Sessions*, pages 69–72, Sydney.

Blaheta, D. and M. Johnson. 2001. Unsupervised learning of multi-word verbs. In *Proceedings of the ACL/EACL 2001 Workshop on the Computational Extraction, Analysis and Exploitation of Collocations*, pages 54–60.

Brants, T. and A. Franz. 2006. {Web 1T 5-gram Version 1}.

Burrows, J. 2002. Delta: A measure of stylistic difference and a guide to likely authorship. *Literary and Linguistic Computing*, 17(3):267–287.

Camacho-Collados, J., M. T. Pilehvar, and R. Navigli. 2015. NASARI: A novel approach to a semantically-aware representation of items. In *Proceedings of NAACL*, pages 567–577, Denver, CO.

Carpineto, C. and G. Romano. 2012. A survey of automatic query expansion in information retrieval. *ACM Computing Surveys (CSUR)*, 44(1):1–50.

Chan, Y. S. and H. T. Ng. 2005. Scaling up word sense disambiguation via parallel texts. In *AAAI*, volume 5, pages 1037–1042.

Chaplot, D. S., P. Bhattacharyya, and A. Paranjape. 2015. Unsupervised word sense disambiguation using Markov random field and dependency parser. In *AAAI*, pages 2217–2223.

Church, K. W. and P. Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16(1):22–29.

Cong, J. and H. Liu. 2014. Approaching human language with complex networks. *Physics of life Reviews*, 11(4):598–618.

Curran, J. R, T. Murphy, and B. Scholz. 2007. Minimising semantic drift with mutual exclusion bootstrapping. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, volume 6, pages 172–180, Bali.

Dagan, I. and O. Glickman. 2004. Probabilistic textual entailment: Generic applied modeling of language variability. *Learning Methods for Text Understanding and Mining*, pages 26–29, Grenoble.

Dang, H. T. 1975. *Investigations into the Role of Lexical Semantics in Word Sense Disambiguation*. Ph.D. thesis, University of Pensylvania.

De Cao, D., R. Basili, M. Luciani, F. Mesiano, and R. Rossi. 2010. Robust and efficient page rank for word sense disambiguation. In *Proceedings of the 2010 Workshop on Graph-based Methods for Natural Language Processing*, pages 24–32, Uppsala.

DeGroot, M. H., M. J. Schervish, X. Fang, Ligang L., and D. Li. 1986. *Probability and Statistics*, volume 2. Addison-Wesley, Reading, MA.

Dice, L. R. 1945. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302.

Erdem, A. and M. Pelillo. 2012. Graph transduction as a noncooperative game. *Neural Computation*, 24(3):700–723.

Evert, S. 2008. Corpora and collocations. *Corpus Linguistics. An International Handbook*, 2:223–233.

Firth, J. R. 1957. A synopsis of linguistic theory 1930–1955. *Studies in Linguistic Analysis*, Oxford: Blackwell.

Fkih, F. and M. N. Omri. 2012. Learning the size of the sliding window for the collocations extraction: A ROC-based approach. In *Proceedings of the 2012 International Conference on Artificial Intelligence (ICAI'12)*, pages 1071–1077.

Gale, W. A., K. W. Church, and D. Yarowsky. 1992. A method for disambiguating word senses in a large corpus. *Computers and the Humanities*, 26(5-6):415–439.

Haveliwala, T. H. 2002. Topic-sensitive pagerank. In *Proceedings of the 11th International Conference on the World Wide Web*, pages 517–526. Maui, HI.

Hoffart, J., S. Seufert, D. B. Nguyen, M. Theobald, and G. Weikum. 2012. KORE: Keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 545–554.

Hoffart, J., F. M. Suchanek, K. Berberich, and G. Weikum. 2013. YAGO2: A spatially and temporally enhanced knowledge base from Wikipedia. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, pages 3161–3165, Beijing.

Holland, J. H. 1975. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*. University of Michigan Press.

Hummel, R. A. and S. W. Zucker. 1983. On the foundations of relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (3):267–287.

Jiang, J. J. and D. W Conrath. 1997. Semantic similarity based on corpus statistics and lexical taxonomy. In *International Conference Research on Computational Linguistics*, pages 970–985.

Jordan, M. I. and Y. Weiss. 2002. Graphical models: Probabilistic inference. In M. A. Arbib, editor, *The Handbook of Brain Theory and Neural Networks*. MIT Press, pages 490–496.

Kelly, E. F. and P. J. Stone. 1975. *Computer Recognition of English Word Senses*, volume 13. North-Holland.

Kilgarriff, A. 1997. I don't believe in word senses. *Computers and the Humanities*, 31(2):91–113.

Kitamura, W. and Y. Matsumoto. 1996. Automatic extraction of word sequence correspondences in parallel corpora. In *Proceedings of the 4th Workshop on Very Large Corpora*, pages 79–87, Copenhagen.

Kleinberg, J. and E. Tardos. 2002. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. *Journal of the ACM (JACM)*, 49(5):616–639.

Koeling, R., D. McCarthy, and J. Carroll. 2005. Domain-specific sense distributions and predominant sense acquisition. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 419–426, Vancouver.

Lafon, P. 1980. Sur la variabilité de la fréquence des formes dans un corpus. *Mots*, 1(1):127–165.

Larsen-Freeman, D. and L. Cameron. 2008. *Complex Systems and Applied Linguistics*. Oxford University Press.

Leech, G. 1992. 100 million words of English: The British National Corpus (BNC). *Language Research*, 28(1):1–13.

Lesk, Michael. 1986. Automatic sense disambiguation using machine readable dictionaries: How to tell a pine cone from an ice cream cone. In *Proceedings of the 5th Annual International Conference on Systems Documentation*, pages 24–26, Toronto.

Leyton-Brown, K. and Y. Shoham. 2008. Essentials of game theory: A concise multidisciplinary introduction. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 2(1):1–88.

Manion, S. L. and R. Sainudiin. 2014. An iterative sudoku style approach to subgraph-based word sense disambiguation. In *Proceedings of the Third Joint Conference on Lexical and Computational Semantics (* SEM 2014)*, pages 40–50, Dublin.

McCarthy, D., R. Koeling, J. Weeds, and J. Carroll. 2007. Unsupervised acquisition of predominant word senses. *Computational Linguistics*, 33(4):553–590.

Menai, M. 2014. Word sense disambiguation using evolutionary algorithms—Application to Arabic language. *Computers in Human Behavior*, 41:92–103.

Mihalcea, R. 2005. Unsupervised large-vocabulary word sense disambiguation with graph-based algorithms for sequence data labeling. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 411–418, Vancouver.

Mihalcea, R. 2006. Knowledge-based methods for WSD. In E. Agirre and P. Edmonds, editors, *Word Sense Disambiguation: Algorithms and Applications*. Springer, pages 107–131.

Mihalcea, R. and D. I. Moldovan. 2001. eXtended WordNet: progress report. In *Proceedings of NAACL Workshop on WordNet and Other Lexical Resources*, pages 95–100, Pittsburgh, PA.

Mihalcea, R., P. Tarau, and E. Figa. 2004. Pagerank on semantic networks, with application to word sense disambiguation. In *Proceedings of the 20th International Conference on Computational Linguistics*, page 1126, Geneva.

Miller, D. A. and S. W. Zucker. 1991. Copositive-plus Lemke algorithm solves polymatrix games. *Operations Research Letters*, 10(5):285–290.

Miller, G. A. 1995. WordNet: A lexical database for English. *Communications of the ACM*, 38(11):39–41.

Miller, G. A., C. Leacock, R. Tengi, and R. T. Bunker. 1993. A semantic concordance. In *Proceedings of the Workshop on Human Language Technology*, pages 303–308, Princeton, NJ.

Moro, A. and R. Navigli. 2015. SemEval-2015 Task 13: Multilingual All-Words Sense Disambiguation and Entity Linking. In *Proceedings of SemEval-2015*, pages 288–297.

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

Moscato, P. 1989. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech Concurrent Computation Program, C3P Report*, 826:1989.

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

Navigli, R. 2006. Meaningful clustering of senses helps boost word sense disambiguation performance. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 105–112, Sydney.

Navigli, R. 2009. Word sense disambiguation: A survey. *ACM Computing Surveys (CSUR)*, 41(2):10.

Navigli, R., D. Jurgens, and D. Vannella. 2013. Semeval-2013 task 12: Multilingual word sense disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (* SEM)*, volume 2, pages 222–231.

Navigli, R. and M. Lapata. 2007. Graph connectivity measures for unsupervised word sense disambiguation. In *IJCAI*, pages 1683–1688.

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

Navigli, R., K. C. Litkowski, and O. Hargraves. 2007. SemEval-2007 task 07: Coarse-grained English all-words task. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 30–35, Prague.

Navigli, R. and S. P. Ponzetto. 2012a. BabelNet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250.

Navigli, R. and S. P. Ponzetto. 2012b. Joining forces pays off: Multilingual joint word sense disambiguation. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1399–1410, Jeju Island.

Navigli, R. and P. Velardi. 2005. Structural semantic interconnections: a knowledge-based approach to word sense disambiguation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(7):1075–1086.

Ng, H. T. and H. B. Lee. 1996. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 40–47, Santa Cruz, CA.

Nowak, M. A, N. L. Komarova, and P. Niyogi. 2001. Evolution of universal grammar. *Science*, 291(5501):114–118.

Okasha, Samir and Ken Binmore. 2012. *Evolution and Rationality: Decisions, Co-operation and Strategic Behaviour*. Cambridge University Press.

Page, L., S. Brin, R. Motwani, and T. Winograd. 1999. The PageRank citation ranking: Bringing order to the Web, Technical Report. Stanford Infolab.

Palmer, M., C. Fellbaum, S. Cotton, L. Delfs, and H. T. Dang. 2001. English tasks: All-words and verb lexical sample. In *Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 21–24, Toulouse.

Pantel, P. and D. Lin. 2002. Discovering word senses from text. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 613–619, Edmonton.

Patwardhan, S. and T. Pedersen. 2006. Using WordNet-based context vectors to estimate the semantic relatedness of concepts. In *Proceedings of the EACL 2006 Workshop Making Sense of Sense-Bringing Computational Linguistics and Psycholinguistics Together*, volume 1501, pages 1–8, Trento.

Patwardhan, Siddharth, Satanjeev Banerjee, and T. Pedersen. 2003. Using measures of semantic relatedness for word sense disambiguation. In *Computational Linguistics and Intelligent Text Processing*, pages 241–257.

Pedersen, T. 2012. Duluth: Measuring degrees of relational similarity with the gloss vector measure of semantic relatedness. In *First Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 497–501.

Pelillo, M. 1997. The dynamics of nonlinear relaxation labeling processes. *Journal of Mathematical Imaging and Vision*, 7(4):309–323.

Pham, T. P., H. T. Ng, and W. S. Lee. 2005. Word sense disambiguation with semi-supervised learning. In *Proceedings of the National Conference on Artificial Intelligence*, volume 20, pages 1093–1098, Ann Arbor, MI.

Pietarinen, A. (editor). 2007. *Game Theory and Linguistic Meaning*. Elsevier.

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

Ponzetto, S. P. and R. Navigli. 2010. Knowledge-rich word sense disambiguation rivaling supervised systems. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1522–1531, Uppsala.

Pradhan, S. S., E. Loper, D. Dligach, and M. Palmer. 2007. SemEval-2007 task 17: English lexical sample, SRL and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 87–92, Prague.

Rao, C. R. 2002. Karl Pearson chi-square test—the dawn of statistical inference. In C. Huber-Carol et al., editors, *Goodness-of-fit Tests and Model Validity*. Springer, pages 9–24.

Rentoumi, V., G. Giannakopoulos, V. Karkaletsis, and G. A. Vouros. 2009. Sentiment analysis of figurative language using a word sense disambiguation approach. In *Proceedings of RANLP*, pages 370-375.

Resnik, P. 1995. Using information content to evaluate semantic similarity in a taxonomy. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 1*, pages 448–453, Montreal.

Rota Buló, S., M. Pelillo, and I. M. Bomze. 2011. Graph-based quadratic optimization: A fast evolutionary approach. *Computer Vision and Image Understanding*, 115(7):984–995.

Sandholm, W. H. 2010. *Population Games and Evolutionary Dynamics*. MIT Press.

Schwab, Didier, Andon Tchechmedjiev, Jérôme Goulian, Mohammad Nasiruddin, Gilles Sérasset, and Hervé Blanchon. 2013. GETALP System: Propagation of a Lesk Measure through an Ant Colony Algorithm. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 232–240, Atlanta, GA.

Sinha, R. Som and R. Mihalcea. 2007. Unsupervised graph-based word sense disambiguation using measures of word semantic similarity. In *Proceedings of ICSC*, volume 7, pages 363–369.

Skyrms, B. 2010. *Signals: Evolution, Learning, and Information*. Oxford University Press.

Smith, J. M. and G. R. Price. 1973. The logic of animal conflict. *Nature*, 246:15.

Smrž, P. 2006. Using WordNet for opinion mining. In *Proceedings of the Third International WordNet Conference*, pages 333–335, Jeju Island.

Snyder, B. and M. Palmer. 2004. The English all-words task. In *Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pages 41–43.

Søgaard, A., A. Johannsen, B. Plank, D. Hovy, and H. M. Alonso. 2014. What's in a p-value in NLP? In *Proceedings of the Eighteenth Conference on Computational Natural Language Learning (CONLL14)*, pages 1–10, Baltimore, MD.

Steinwart, I. and A. Christmann. 2008. *Support Vector Machines*. Springer Science.

Strapparava, C., A. Gliozzo, and C. Giuliano. 2004. Pattern abstraction and term similarity for word sense disambiguation: Irst at Senseval-3. In *Proceedings of SENSEVAL-3 Third International Workshop on Evaluation of Systems for the Semantic Analysis of Text*, pages 229–234.

Szabó, G. and G. Fath. 2007. Evolutionary games on graphs. *Physics Reports*, 446(4):97–216.

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

Tong, H., C. Faloutsos, and J. Pan. 2006. Fast random walk with restart and its applications. In *Proceedings of the Sixth International Conference on Data Mining*, pages 613–622, Hong Kong.

Tratz, S., A. Sanfilippo, M. Gregory, A. Chappell, C. Posse, and P. Whitney. 2007. PNNL: A supervised maximum entropy approach to word sense disambiguation. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 264–267, Prague.

Tripodi, R. and M. Pelillo. 2015. WSD-games: A game-theoretic algorithm for unsupervised word sense disambiguation. In *Proceedings of SemEval-2015*, pages 329–334, Denver, CO.

Tripodi, R., M. Pelillo, and Rodolfo Delmonte. 2015. An evolutionary game theoretic approach to word sense disambiguation. In *Proceedings of Natural Language Processing and Cognitive Science 2014*, pages 39–48, Venice.

Tsang, E. 1995. *Foundations of Constraint Satisfaction*. Academic Press.

Véronis, J. 2004. Hyperlex: lexical cartography for information retrieval. *Computer Speech & Language*, 18(3):223–252.

Vickrey, D., L. Biewald, M. Teyssier, and D. Koller. 2005. Word-sense disambiguation for machine translation. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 771–778, Vancouver.

Von Neumann, J. and O. Morgenstern. 1944. *Theory of Games and Economic Behavior*. Princeton University Press.

Weaver, W. 1955. Translation. *Machine Translation of Languages*, 14:15-23.

Weibull, J. W. 1997. *Evolutionary Game Theory*. MIT Press.

Wu, Z. and M. Palmer. 1994. Verbs, semantics, and lexical selection. In

Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, pages 133–138, Las Cruces, NM.

Yarowsky, D. 1993. One sense per collocation. In *Proceedings of the Workshop on Human Language Technology*, pages 266–271, Princeton, NJ.

Yarowsky, D. and R. Florian. 2002. Evaluating sense disambiguation across diverse parameter spaces. *Natural Language Engineering*, 8(04):293–310.

Zhong, Z. and H. T. 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.

Zhong, Z. and H. T. Ng. 2012. Word sense disambiguation improves information retrieval. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 273–282, Jeju Island.