Scuola Dottorale di Ateneo
Graduate School

Dottorato di ricerca
in Informatica
Ciclo XXVIII
Anno di discussione 2016

Evolutionary Game Theoretic Models for Natural Language Processing

Settore scientifico disciplinare di afferenza: INF/01
Tesi di Dottorato di Rocco Tripodi, matricola 813696

Coordinatore del Dottorato	Tutore del Dottorando
Prof. Riccardo Focardi	Prof. Marcello Pelillo

Università Ca' Foscari di Venezia
Dottorato di Ricerca in Informatica, XXVIII ciclo

Ph.D. Thesis

Evolutionary Game Theoretic Models for Natural Language Processing

Rocco Tripodi

Supervisor
Marcello Pelillo

PhD Coordinator
Riccardo Focardi

Referee
–

Referee
–

Author's Web Page: http://dais.unive.it/~tripodi

Author's e-mail: rocco.tripodi@unive.it

Author's address:

Dipartimento di Scienze Ambientali, Informatica e Statistica
Università Ca' Foscari di Venezia
Via Torino, 155
30172 Venezia Mestre – Italia
tel. +39 041 2348411
fax. +39 041 2348419
web: http://www.dsi.unive.it

♓

Abstract

This thesis is aimed at discovering new learning algorithms inspired by principles of biological evolution, which are able to exploit relational and contextual information, viewing clustering and classification problems in a dynamical system perspective. In particular, we have investigated how game theoretic models can be used to solve different Natural Language Processing tasks. Traditional studies of language have used a game-theoretic perspective to study how language evolves over time and how it emerges in a community but to the best of our knowledge, this is the first attempt to use game theory to solve specific problems in this area.

These models are based on the concept of equilibrium, a state of a system, which emerges after a series of interactions among the elements, which are part of it. Starting from a situation in which there is uncertainty about a particular phenomenon, they describe how a disequilibrium state resolves in equilibrium. The games are situations in which a group of objects has to be classified or clustered and each of them has to choose its collocation in a predefined set of classes. The choice of each one is influenced by the choices of the other and the satisfaction that a player has, about the outcome of a game, is determined by a payoff function, which the players try to maximize. After a series of interactions the players learn to play their best strategies, leading to an equilibrium state and to the resolution of the problem.

From a machine-learning perspective this approach is appealing, because it can be employed as an unsupervised, semi-supervised or supervised learning model. We have used it to resolve the word sense disambiguation problem. We casted this task as a constraint satisfaction problem, where each word to be disambiguated is constrained to choose the most coherent sense among the available, according to the sense that the words around it are choosing. This formulation ensures the maintenance of textual coherence and has been tested against state-of-the-art algorithms with higher and more stable results.

We have also used a game theoretic formulation, to improve the clustering results of dominant set clustering and non-negative matrix factorization technique. We evaluated our system on different document datasets through different approaches, achieving results, which outperform state-of-the-art algorithms.

This work opened new perspectives in game theoretic models, demonstrating that these approaches are promising and that they can be employed also for the resolution of new problems.

Acknowledgments

I would like to thank my supervisor, Prof. Marcello Pelillo for his guidance throughout the course of this work, Prof. Rodolfo Delmonte for his suggestions and Prof. Bernadette Sharp for her kind help during my visit in Stafford. I would like to thank my family for their unrestricted support and for giving me the possibility to became who I am. And a special thanks to Luana, with whom we are playing the games and to my sister with whom we share a lot of similarities.

Contents

List of Figures

List of Tables

Introduction

I.1 Learning Games

This thesis is aimed at discovering new learning algorithms inspired by principles of biological evolution, which are able to exploit relational and contextual information, viewing clustering and classification problems in a dynamical system perspective. In particular, we have investigated how game theoretic models can be used to solve different Natural Language Processing (NLP) tasks. Traditional studies of language have used a game-theoretic perspective to investigate how language evolves over time and how it emerges in a community of speakers, but to the best of our knowledge, this is the first attempt to employ a game-theoretic perspective to solve specific problems in the area of NLP.

These models are centered around the concept of *equilibrium*, which is reached after a series of interactions among the elements which are part of the observed phenomenon. Starting from a situation in which there is confusion or uncertainty about a particular problem or situation, they model how a system evolves over time and describe how a state of disequilibrium resolves in equilibrium.

We modelled the games as situations in which there is a set of objects, which have to be classified or clustered. Each object is seen as a player of the games, which has to choose its class membership, among a predefined number of classes. The choice of each one is influenced by the choices of the others and the satisfaction that a player has, about the outcome of a game, is encoded in a payoff function, which the players try to maximize, in order to *win* the games. After a series of interactions, the players learn to play their best strategies, leading to an equilibrium state, in which no one can improve its condition, unilaterally changing its strategy. This equilibrium state of the system leads to the resolution of the problem, where each object is placed in a category or cluster.

We interpret these tasks as games, in which each object to be clustered or classified is a member of a population of players, which play the games with a certain number of other players. With this interpretation, players have a predetermined number of actions, which they can take. These actions have a direct correspondence with the classes or clusters, which each player has to choose, in order to solve the task. The players play the games repeatedly, adapting their choices according to what has been effective and what has not in previous games. The changes in the population's decisions are the result of natural selection, which is used to drive the choices of the players, indicating the best strategy to adopt[1]. Once the equilibrium

[1]Natural selection is the process that allows determined phenotypes (traits or behaviors) to

is reached, it is possible to associate each object to one class, if it is required an hard partitioning of the data, or it is possible to associate each object to more classes, where each object has a probability distribution over them, if it is required a soft classification.

Defining the problem in this way has many advantages. First of all, it offers the advantage of biological plausibility, showing for each object the patters of evolution and identifying clearly what are the variables, which lead to determined states. A feature which is missing in many approaches, such as *deep learning*, for example, in which numerous hyper-parameters have to be trained carefully and whose performances are justified empirically rather than theoretically [1]. Furthermore, these models have a solid mathematical foundation and provide a set of powerful and elegant differential equations,which are able to find equilibrium states, as discovered, in classical game theory, by John Nash, in 1951 [2] and extended to evolutionary games by John Maynard Smith and George Price, in 1973 [3].

From a machine-learning perspective this approach is appealing, because it can be employed as an unsupervised, semi-supervised or supervised learning model. An unsupervised learning model can be developed, formulating the games in a way in which, at the beginning, the system has only players with uncertainty about the strategy to employ, usually, this situation is described by a uniform distribution over the players' strategies. The payoff function of the games is encoded in terms of contextual and relational information, which enables players to adjust (learn) their strategies. A semi-supervised setting can be employed, forcing the players about which it is known their class, to always play the games with a defined strategy, without uncertain, and the others players, which have not defined their strategy, to be initialized with a uniform distribution over their strategy. In this case the players with a defined strategy will influence other players, which gradually will learn to play according to the strategy of the *labeled* players. A supervised setting can be used to train a classifier based on the information derived from previous equilibrium states.

The most important part of these models is the calculation of the payoff function of the games, which has to take into account relational and contextual information, in order to supply the right feedback to the players. The relational information is used to model the geometry of the data. The system is described as a weighted graph, $G = (V, E, w)$, whose vertices, V are the objects to be classified, the edges, $E \subseteq V \times V$, indicate interactions among the players and the weights, $w : E \to \mathbb{R}$, indicate the pairwise similarity among the players.

The pairwise similarity information is used to balance the reciprocal influence among the players during the interactions. Instead, the contextual information is encoded as class similarity function. We employ an ontological representation of the classes, which allows us to structure the classes and to find correlations among them. In this way it is possible to overcome the limitation of the *homophily principle*, in

reproduce more than others. These phenotypes have the possibility to transmit their traits to the feature generations and to survive in determined environments.

which similar objects have to be classified in the same class [4]. Instead, we propose that similar objects should be classified in similar classes, because, especially, in fine-grained classification tasks, it is possible that two objects, which enjoy a strong pairwise similarity, have to be classified in two distinct classes, which in many cases are contiguous categories in an ontology or in a taxonomy[2].

In this work we have employed an evolutionary game theoretic perspective to resolve the word sense disambiguation task, which consists in finding the appropriate sense for all the words in a text. This problem is particularly challenging, because many words in the vocabulary of a language are ambiguous and have multiple meanings, depending on the context in which they are used. For example, the word *star* can refer to a *celestial body*, in the field of astronomy, or to a *celebrity*, in popular culture.

We casted this task as a constraint satisfaction problem, where each word to be disambiguated is constrained to choose the most coherent meaning among the available, given the meaning that the other words around it are going to choose. This formulation ensures the maintenance of the textual coherence, taking into account the overall meaning of the analyzed text, a feature which is missing in many state-of-the-art systems. We tested our approach in two modalities, unsupervised and semi-supervised and evaluated them against state-of-the-art algorithms. The results of this evaluation show that our approach has higher and more stable performances, in terms of precision and recall.

The last part of this work is devoted to document clustering. We have employed a game-theoretic perspective to improve the clustering results of the *dominant set* clustering algorithm [5]. In this is field, to the best of our knowledge, this algorithm has never been tested. We used the dominant set algorithm to obtain small clusters of objects and used this information to initialize the strategy space of these players, leaving the other players with a uniform distribution over their strategies. In this way the information about the strategy choice of clustered players is conveyed to players which have not employed a defined strategy, yet. We evaluated this approach on twelve datasets, using different approaches for the construction of the similarity graphs, obtaining good results. We also tested dominant set, in a setting, in which the number of clusters to extract is not given in advance. This information is required by many clustering algorithms but is hard to obtain in real-life applications. In this way we tested the ability of our algorithm to find *natural clusters*, which is desirable in many real applications.

A similar perspective has been used to to refine the clustering results obtained with the non-negative matrix factorization technique [6], This work consists in using the results obtained with this technique to define the inclination that each document has toward a determined cluster. After this system initialization, we started the dynamics of the document clustering games, assigning each document to a cluster. Each time this technique has been employed, it was possible to obtain higher results

[2]For example, two animals which are subspecies of the same species

than those provided initially by the non-negative matrix factorization.

The difference among the two approaches, dominant set and non-negative matrix factorization, lies in the fact that with the dominant set we obtain a hard partition of the players, and with non-negative matrix factorization, we obtain the propensity that the players have toward a particular strategy. With the dominant set, the players are: clustered, choosing always a determined strategy, which can not be changed; or unclustered, players with maximum grade of uncertainty. With non-negative matrix factorization, we obtain the inclination that each player has toward its strategies, a situation, which can be modified according to what neighbors players do.

In the following three sections we introduce some related work on the application of game theory to study different aspects of language, some concepts to support the interpretation of language as a complex adaptive system and some theories of learning in games. The thesis continues with an introduction to game theory and evolutionary game theory, In Chapter 1. In Chapter 2 it is described our game theoretic approach to word sense disambiguation. Finally, in Chapter 3 it is proposed our approach to document clustering.

The overall contribution of this thesis is that it shows how game-theoretic models can be employed for different NLP tasks. We have opened a new perspective in this discipline, exploring different techniques in game theory for classification and clustering, demonstrating that these techniques can have a large number of applications arising from a variety of fields.

I.2 Language and Games

The idea of using principles derived from Game Theory to NLP tasks came to mind when we considered the historical relation that there is among the concept of *game* and the act of communicating through a common language. The so-called *game metaphor* has been used in different ways by philosophers and linguists to explain how language has been developed and how it works.

During the XIX century, Charles Sanders Peirce (renowned for the introduction of abductive reasoning in logic), and Ferdinand de Saussure (the father of structural linguistics), independently, likened language to chess. Peirce asserted that expressions mediate thoughts just as pawns and knights mediate the strategy intentions of a chess player [7]. De Saussure affirmed that the two activities, language and chess, both involve dynamics, conventional rules, and positional strategies [8]. The analogy, proposed by de Saussure, describes language as a dynamical system (that is a system which evolves over time), whose properties are convectional and emerge from a long series of interactions among the speakers.

A similar perspective, for the origin of language, was proposed even earlier, by Diodorus of Sicily (90-27 BC). He described this phenomenon, considering also its probabilistic and dynamic nature, as follows:

> The sounds they made had no sense and were confused; but gradually they articulated their expressions, and by establishing symbols among themselves for every sort of object they came to express themselves on all matters in a way intelligible to one another. Such groups came into existence throughout the inhabited world, and not all men had the same language, since each group organized their expressions as chance had it."

In this description language is seen as a human construction, in which from arbitrary meaningless sounds, it is possible to gradually converge toward an equilibrium in which the sounds acquire a common meaning in a population. It is interesting to note the similarity that this perspective has with the considerations given by Darwin, nineteen centuries later, on the same topic:

> The formation of different languages and of distinct species, and the proofs that both have been developed through a gradual process, are curiously parallel. [...] Max Muller has well remarked: 'A struggle for life is constantly going on amongst the words and grammatical forms in each language. The better, the shorter, the easier forms are constantly gaining the upper hand, and they owe their success to their inherent virtue.' To these important causes of the survival of certain words, mere novelty and fashion may be added; for there is in the mind of man a strong love for slight changes in all things. The survival or preservation of certain favoured words in the struggle for existence is natural selection." [9]

This idea of the spontaneous formation of a common language, which gradually emerges from the interactions among the speakers, was reconsidered and systematized in the XX century, with the work of Ludwig Wittgenstein, one of the most prominent philosophers of his time. He introduced the concept of *language game*, which can be considered as the theoretical foundation of many computational models of language. The basic idea of *language games* can be encapsulated in this concise statement:

> [...] the meaning of a word is its use in the language. [10]

From this perspective, the meaning of a word is not predefined, it depends on how speakers use words in specific contexts. In fact, using a word in actual situations gives rise to correlations among words and objects. The words' meaning is constructed by virtue of these repeated actions, which show the conventional nature of language.

The recognition of these affinities is the starting point for the interpretation of language as a dynamical system, but, all these intuitions were not developed into a general theory, remaining fragmented, until the second half of the XX century. This essentially is for two reasons:

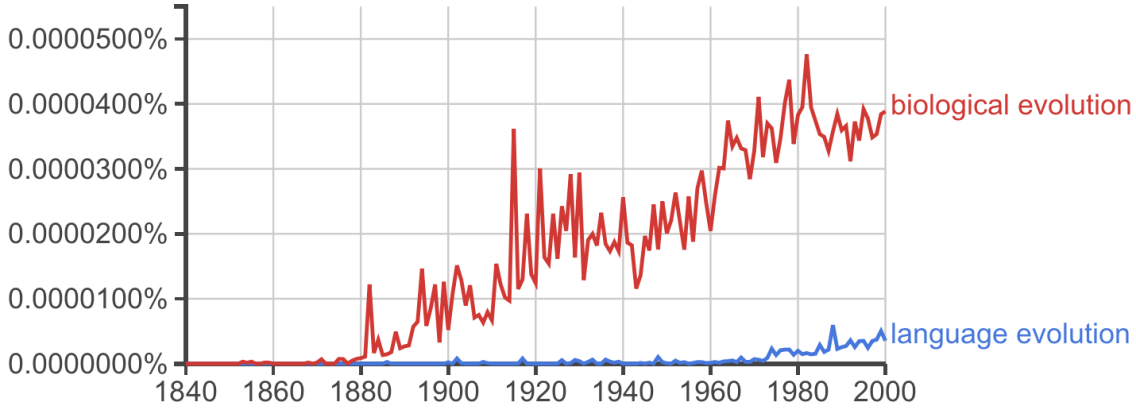


Figure 1: On the vertical axis it is indicated the frequency of the two expressions: *biological evolution* and *language evolution*, in the English scientific literature, as collected by Google. The year of measurement are indicated on the horizontal axis.

1. the lack of theoretical models which explain how evolutionary processes work.

2. the lack of empirical tools, able to simulate evolutionary processes.

A theoretical model for evolutionary processes was given in 1859, when Charles Darwin published *The Origin of Species*, but the studies on language evolution did not start soon after this important contribution, due to the ban that the *Sociétéde Linguistique de Paris* imposed on this topic. As we can see, from Figure 1[3], the scientific community started to talk about this concept during the XX century. It increased his popularity after 1960, following a similar trend to that for the broader discipline of *biological evolution*. Within a Darwinian framework, the language can be interpreted as an evolving system. In this system the features of the language that are more efficient, from a communicative perspective (easy to learn and to use), will tend to spread in the speakers' population, whereas inefficient features will tend to disappear [12]. This phenomenon can be explained considering that speakers with good communications skills are favoured by natural selection and are more likely to reproduce than others [13].

Now that the theoretical model, which explains how evolutionary processes take place, had been given by Darwin, what remained to discover was an empirical model for the study of evolutionary dynamics. Even if, methods for the study of non-linear differential equations, the essential tools for the analysis of dynamical systems, date back to 1890, when Poincaré published a work on celestial mechanics [14], these kinds of tools have not been used to study biological or social phenomena, such as the evolution of language. Instead, they were used to model phenomena in physics,

[3]The figure has been obtained using Google N-gram viewer. We refer to [11] for a complete description of the tool and the data used. The viewer is accessible at https://books.google.com/ngrams/ .

chemistry, economy and population genetics. It is only in 1963, with the work of John Maynard Smith and George R. Price, that a framework was introduced to model evolutionary processes in biology [3]. More specifically, they developed a framework for understanding ritualized behaviours in animal conflicts. This work merged the Darwinist idea of evolution and Game Theory, giving rise to the emergence of Evolutionary Game Theory.

The core idea, of the framework introduced by Smith and Price, is that of thinking evolution taking into account how determined strategies (phenotypes) have the ability to prevail over others and to reproduce themselves in the population. In this context, individuals can dynamically adapt their strategies to the environment, in order to survive in the population.

Smith and Price introduced the concept of Evolutionary Stable Strategy (ESS), to determine how the strategies of a population evolve over time. This concept can be considered as equivalent to the concept of Nash Equilibrium in non-cooperative games (see Section 1 for a more detailed introduction to Game Theory). In non-cooperative games, we have two (or more) players, which have to choose among a set of strategies to be used against their co-player. Each possible combination of strategies, which can be selected by the players, is associated to a payoff, received by the players, when they adopt the corresponding strategies.

A Nash Equilibrium is a state of the game, in which the players have no incentive to deviate from their strategies, because there is no way to obtain a higher payoff, given the strategy of the co-player. The concept of ESS is used to describe a situation, in which, once a population has adopted a determined strategy, it will be maintained by the population, even if a small number of players (mutants) start to play a different strategy, for this reason it is said to be evolutionary stable.

The concept of ESS is important because it explains how ritualized behaviours emerge in a population and how they evolve over time. Within this framework it is possible to develop quantitative simulations that can be tested empirically with high precision.

David Lewis, with the book entitled *Convention* [15], was the first, to use some concepts derived from game theory to explain the emergence of language conventions, such as, the use of a common vocabulary among the speakers of a community. For the first time, the theoretical model proposed by Wittgenstein was converted into a mathematical model. Lewis proposed a basic model to explain a simple signalling system. In its simplest representations, there are two players, the sender and the receiver, and N possible states of the world. The sender observes the state of the world, which is selected randomly by Nature and chooses, among a set of N symbols, the appropriate one to send to the receiver, in order to communicate the corresponding state of the world. The receiver observes the received symbol and selects, among a set of N acts, the most appropriate one to adopt in response to the symbol. In this formulation only one act is correct given a specific state. In this way, sender and receiver have a common interest to coordinate their choices, otherwise it is not possible to create an efficient communication system. In fact,

they both receive a reward (payoff) in the case of correct matching, otherwise they receive nothing. A *signalling system equilibrium* is reached if it is guaranteed that the correct matching, among symbols and states, is always adopted.

In this framework, the sender's strategy maps states to symbols, the receiver's strategy maps symbols to acts. Formally, we can describe the sender's strategy as an $N \times X$ matrix P, where the rows are indexed according to the states and the columns are indexed according to the symbols, where an entry p_{ij} has a value between zero and one, which expresses the probability that symbol j is used to communicate state i and each row sum up to 1. The receiver's matrix, Q, has the same properties as the sender's matrix, the only differences are that its rows are indexed according to the symbols, its columns are indexed according to the acts and its values indicate the probability that the receiver chooses act j in response to symbol i. The payoff of the signalling game can be computed with the following equation:

$$\pi(P, Q) = \frac{1}{N} \sum_{i,j} p_{ij} q_{ji}. \tag{1}$$

A *signalling system equilibrium* is only possible when, in P, each state is mapped to a different symbol, in Q, each symbol is mapped to a different act and $p_{ij} = q_{ji}$. In fact in this case the payoff of the game is 1, according to equation (1), which is the maximal value it can take and corresponds to the *strict Nash equilibrium* of the game [16].

Lewis' model shows the conventional nature of the symbols' meaning, demonstrating that the association symbol-act is arbitrary. In fact, two permutation matrix, P and Q, can have the same payoff. His model is conceived to discover the *equilibria* of a signalling system, with different numbers of players, states, symbols and acts. Lewis used traditional game theory for this purpose, but this can lead to different non-strict equilibria. For example, there are always *completely pooling equilibria*, in which the sender uses always the same symbol and the receiver uses always the same act; or *partial pooling equilibria*, where only the information about some states is pooled. For example, in the game described by the two matrices in (2), the information about state 3 is always transmitted correctly and the information about state 1 and 2 is pooled.

$$P = \begin{pmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & \lambda & 1-\lambda \end{pmatrix}, Q = \begin{pmatrix} \mu & 1-\mu & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} \tag{2}$$

A solution, for the kind of problems described above, could be to study the evolution of a signalling system, not just its possible equilibria. In this way it is possible to discover equilibria, which are evolutionary stable and correspond to *strict Nash equilibria*. Brian Skyrms [17] proposed to apply concepts derived from evolutionary game theory to study Lewis signalling systems[4]. In this perspective there is a pop-

[4]Concepts which were not available to Lewis, since its work has been published before the

ulation of players, with different strategies, which play the games repeatedly, until the system converges. Players, which adopt an effective communication strategy get an higher payoff than others and, in an evolutionary perspective, have an higher probability to reproduce themselves, incrementing the percentage of players, which play an effective strategy. Skyrms [18] shows how this kind of dynamics leads to successful coordination, producing a *signalling system*, which is ruled by natural selection.

In the same vein of the model proposed above, many other models have been proposed to explain how other characteristics of language follow similar dynamics. Nowak [19, 20] proposed a framework for the evolution of a common lexicon in a population, identifying what is the minimum reproductive rate of the words, which are maintained in the lexicon of a language and what is the maximum size of a lexicon. In [21], there were taken into account the grammatical aspects of language evolution, identifying the dynamics, which involve grammatical acquisition. Also aspects of syntax emergence have been studied in [22].

I.3 Language as a Dynamical System

In the previous section we have outlined how the concept of game can be used to explain the language evolution and the underling vision of language as a dynamical system. We have shaped the background for the interpretation of language as an open and continually evolving system. This can also be verified considering how languages such as English or Italian are evolving by virtue of the speakers, which contribute to passing down the language from one generation to another, modifying and adapting it to new needs. Instead, languages, such as Latin, are extinct, because no one actively uses them. In this section we provide a more detailed explanation of how language can be interpreted as a dynamical system, in particular we will discuss the components, which involve language evolution.

The model proposed by Christiansen and Kirby [12], for the evolution of language, is composed of three distinct but interacting adaptive systems, which operate at three different timescales and are: biological evolution, individual learning and cultural transmission (see Figure 2). Biological evolution is the process in which natural selection fosters individuals with good communication skills. It operates at the timescale of a species. Individual learning involves the personal knowledge of a speaker and operates at its timescale. Finally, cultural transmission involves the dynamics, which permit some characteristics of the language to be maintained from one generation to another or to be modified. In this model there is a circular chain of reciprocal influences: biological evolution influences the learning abilities of a species; the learning abilities influence what will be transmitted by the cultural transmission and cultural transmission modifies the abilities landscape governed by natural selection.

introduction of evolutionary game theory, by Smith and Price [3].

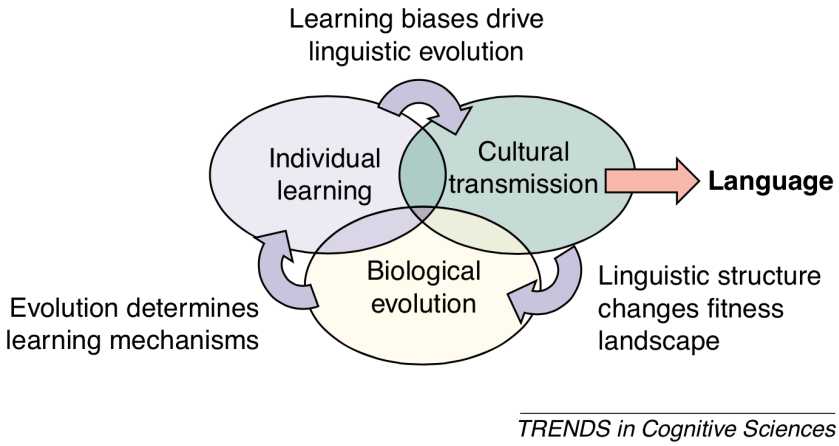


Figure 2: The three adaptive dynamical systems which characterize language evolution.

This perspective is also employed in [23], where it is stated that the interactions among speakers, over a long span of time, can lead to phase transitions in behaviour and linguistic structures. It cites as examples of these phase transitions, the emergence of segmental phonology, the invention of hierarchic morphology and syntax and the use of recursion in sentence construction.

An aspect, which today can be studied, within this perspective, is the lexical diffusion. It refers to the spread of sound changes through the lexicon of a language. This phenomenon can by analysed thanks to the availability of large volume of data, such as the Google N-Grams Corpus [24] or the British National Corpus [25], in which the data are organized temporally.

The studies on this topic are inspired by the work of the Luca Cavalli-Sforza [26], in which the introduction of a new word in the lexicon of a language is interpreted as an analogue of mutation in biology. If observed for a long time, the frequency of an innovation (the new word, in our case), follows an S-shaped curve. At the beginning the frequency rapidly increases, then follows an approximately linear increase, and finally the increase slows.

As an example of this phenomenon, it can be considered the introduction of the word *selfie*, which was used for the first time in 2002, on an Australian Internet forum[5], then, with the help of social media, it rapidly propagates to a larger network, becoming more popular than the replaced word, *self-portrait*. It is also possible to consider the phenomenon of the regularization of verbs, which is similar to the phenomenon described above, but develops on a longer time scale. It has been analyzed in [23] and experimentally verified, using millions of digitalized books, in [27]. The examples of the S-shaped curve, of this phenomenon (which is still

[5]According to The Oxford English Dictionary, which, in 2013, proclaimed *selfie* word of the year.

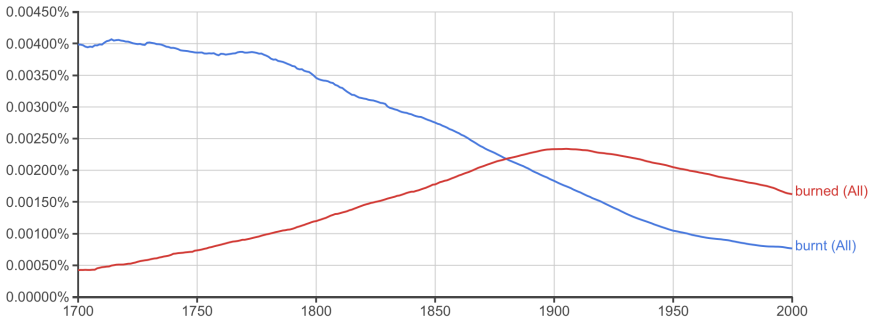


Figure 3: The regularization of verbs *to burn*. The frequency of the word is indicated on the vertical axis, the years of measurements are displayed on the horizontal axes.

evolving), is shown in Figure 3, where we can observe that as the frequency of the irregular form *burnt* decreases, the frequency of the *mutant* form increases.

Phenomena of this kind are produced by the interaction of the speckers in the system of language and are driven by mutation, selection and contagion. They have been studied taking into account the topology of the interaction network, which characterize the system and constructing models, which reflect this structure. In this representation, the agents are represented as nodes on a graph and the interaction are indicated by the presence of an edge connecting two nodes. Binary undirected graphs are the simplest representation of an interaction network. In this representation an edge between two nodes either exists or it does not. Other kinds or representations use weighted graphs, where the weights on the edges denote the similaritiy, the proximity or the level of influence among two nodes.

I.4 Learning in Games

In the context of this work, a learning model can be interpreted as the specification of the learning rules used by individual players to change their behavior, strategies or beliefs, when the games are played repeatedly [28].

Learning theories can be divided into two broad categories, descriptive and prescriptive theories [29, 30]. The former studies show learning occurs, the latter studies what are the instructional methods, which leads to the development of learning abilities. Over the years, this phenomenon has been studied in different ways and from different perspectives. A simple model, which has been proposed by behaviorist psychologists, interprets learning as a passive process, where the learner only responds to environmental stimuli. This interpretation has been replaced, by cognitivist psychologists, with a perspective in which learning is seen as a process of reorganization of the previous knowledge, as new experiences occur. In turn, the cognitivist perspective has been enriched by the social learning paradigm, which interprets learning as a process, in which the reorganization of knowledge is also influenced by success

and failure of other learners.

The most important game theoretic models of learning are: belief learning, reinforcement learning and evolutionary learning [30]. In belief learning the players learn to play their strategies according to the beliefs that they constructed in previous situations. The rule to choose a strategy in this kind of learning is to always play the best response to the strategy played by the co-player in previous periods,

$$P(A) = \frac{w(a)}{\sum_{a' \in A} w(a')}, \tag{3}$$

where A is the co-player's set of possible strategies and $w(a)$ is the number of times the opponent adopted strategy a. This theory has been generalized using a Bayesian updating, for the selection of the best response.

In reinforcement learning, players update their strategies observing the environment and the rewards they receive adopting a determined action. In this case, the decisions, $S = q_i, ..., q_m$ of player n are updated with the following equation,

$$q_{nj} = \begin{cases} q_{nj} + R(x), & \text{if } j = k. \\ q_{nj}, & \text{otherwise.} \end{cases} \tag{4}$$

where $R(x)$ is the reinforcement received playing a determined strategy. The probability of the strategies at time t are,

$$p_{nk} = \frac{q_{nk}(t)}{\sum_{j \in S} q_{nj}(t)}, \tag{5}$$

in this framework the players are considered to behave quite mechanically, simply reacting to positive or negative stimuli [31].

In evolutionary learning, the games are played repeatedly, by a population of agents and the share of population playing each strategy grows at a rate proportional to the payoff obtained playing that particular strategy. Even if this model has been introduced to explain biological evolution, the underlying process, which enables players to adjust their strategies, can be seen as the result of a learning process [32]. The learners receive examples from other players and have to infer the rules that generate their behaviors. The learning process has the form of an inductive inference, after seeing enough examples, the learner can infer the correct strategy to employ.

The equilibrium state of the system can be interpreted as the best configuration that each player can achieve in a particular environment. The environment is composed of a network, which models the interactions among the agents, the knowledge of the set of possible strategies that can be employed and the payoffs, which will be received playing a determined strategy. All this information makes up the background knowledge of the agents, which is given to all of them.

In this context, we can say that the agents are provided with *bounded rationality*, that is to say that their rationality is limited to the information that they have and

that their decisions are made according to this information. *Bounded rationality* has also been used to explain human behavior, in social science, and to model decision-making processes [33].

The main difference among reinforcement learning and evolutionary learning is that evolutionary learning is based on the notion of equilibrium, which is the state of the system in which all players have learned their best strategy, according to what strategy other players have employed. In this case the learning process is shaped at system level and not at single player level. Even if, evolutionary learning can incorporate reinforcement at player level, its result are always based on the aggregated behavior of the populations.

The details of evolutionary learning processes will be given in Chapter 1, where we introduce the replicator dynamics equation, which is used to find the equilibria of the games. It is a powerful tool, which can be used with different payoff functions, depending on the problem at hand and on the process which has to be modeled. It permits the understanding of how each player learns from the others, adjusting its strategy profile until the system converges.

1

Game Theory

1.1 Classical Game Theory

Game theory provides predictive power in interactive decision situations. It was introduced by Von Neumann and Morgenstern [34] 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 used in this thesis) 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 employ in a game.

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

$$u_i(s_i^*, s_{-i}) > u_i(s_i, s_{-i}), \forall s_{-i} \in S_{-i} \tag{1.1}$$

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

As an example, we can consider the famous *Prisoner's Dilemma*. In this game two people have been arrested for a crime committed together. Each person is questioned in a separate room, they cannot communicate but they know that they receive the same proposal: if they both confess they will be jailed for five years, if they both do not confess, they will be jailed for one year and if one confesses and the other does not confess, the one which confesses will be set free and the other will be jailed for six years.

The payoff matrix of this game is shown in Table 1.1, where each cell represents a strategy profile, the first number indicates the payoff of *Player 1* (P_1) and the second is the payoff of *Player 2* (P_2), when both players employ the strategy associated to a specific cell. P_1 is called the *row player* because it selects its strategy according to the rows of the payoff matrix, 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

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

Table 1.1: The payoff matrix of The Prisoner's Dilemma game.

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 his decision, because there is no way to do better. In fact, in the example of *The Prosoner's Dilemma*, for both players, -5 is better than -6 and 0 is better than -1, so *confess* is the best strategy, whatever other strategy the co-player employs.

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 profile 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. Each player i has a strategy profile which is defined as a standard simplex,

$$\Delta = \left\{ x \in \mathbb{R}^n : \sum_{h=1}^{m} x_h = 1, \text{ and } x_h \geq 0 \text{ for all } h \in x \right\} \tag{1.2}$$

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

In a *two-player game*, a strategy profile can be defined 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:

$$u_i(p, q) = p \cdot Aq \ , \ u_j(p, q) = q \cdot A^T p \tag{1.3}$$

where A is the payoff matrix of the game played by i and j and it is assumed to be simmetric. The Nash equilibrium is computed in mixed strategies in the same way of 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.

A game theoretic framework can be considered as a solid tool in decision making situations since a fundamental theorem by Nash [2] states that any normal-form game has at least one mixed Nash equilibrium, which can be employed as the solution of the decision problem.

1.2 Evolutionary Game Theory

Evolutionary game theory was introduced by John Maynard Smith and George Price [3], 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 [35]. It has been introduced in biology to explain the ritualized behaviors which emerge in animal conflicts [3]. In particular, Smith and Price have focused their research on why animals adopt determined strategies instead of others, when they are involved in a conflict with other animals. The study was conducted on species with different characteristics, For example, Smith and Price studied why animals, possessing offensive weapons, do not always use an offensive strategy, which is able to serious injuries others animals,. The answer to this question is that a *total war* strategy is risky and can lead to the extinction of the species. The payoff connected to this strategy is low, because the animal involved in an offensive context can be damaged by the opponent. Furthermore, it is not evolutionarily stable, since members of the population adopting a different strategy can live longer.

In this context, strategies correspond to phenotypes (traits or behaviors), payoffs correspond to offspring, allowing players with a high actual payoff (obtained thanks to its phenotype) to be more prevalent in the population. This formulation explains natural selection choices between alternative phenotypes based on their utility function. This aspect can be linked to rational choice theory, in which players make a choice that maximizes its utility, balancing cost against benefits [36].

This intuition introduces an *inductive learning* process, in which we have a population of agents which 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. It lives in the mixed strategy space of the game, which is given by the Cartesian product:

$$\Theta = \times_{i \in I} \Delta_i \tag{1.4}$$

The expected payoff of a pure strategy e^h in a single game is calculated as in mixed strategies (see Equation 1.3). The difference in evolutionary game theory is that a player can play the games with all other players, obtaining a final payoff which is the sum of all the partial payoffs obtained during the single games. The payoff corresponding to a single strategy can be computed as:

$$u_i(e_i^h) = \sum_{j=1}^{n} (A_{ij} x_j)_h \tag{1.5}$$

and the average payoff is:

$$u_i(x) = \sum_{j=1}^{n} x_i^T A_{ij} x_j \tag{1.6}$$

where n is the number of players with whom the games are played and A_{ij} is the payoff matrix among player 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 his strategy space according to the payoffs gained during the games, allowing the player to allocate more probability on the strategies with high payoff, until an equilibrium is reached, which means that the strategy spaces of the players cannot be updated, because it is not possible to obtain higher payoffs. The replicator dynamic equation [37] is used In order to find those states, which correspond to the Nash equilibria of the games,:

$$\dot{x} = [u(e^h, x) - u(x, x)] \cdot x^h \forall h \in S \tag{1.7}$$

This equation allows better than average strategies (best replies) to grow at each iteration. It can be used as a tool in dynamical systems to analyze frequency-dependent selection [32]. It assumes that the grow rate of each strategy is proportional to its fitness, which is defined summing the payoffs gained during the games. In this context, several strategies can coexist with different rates.

The following theorem states that the fixed points of equation 1.7 are Nash equilibria.

Theorem 1. *A point $x \in \Theta$ is the limit of a trajectory of equation 1.7 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.*

Proof. See Weibull [38].

For the experiments of this thesis the discrete time version of the replicator dynamic equation was used:

$$x^h(t+1) = x^h(t) \frac{u(e^h, x)}{u(x, x)} \ \forall h \in S \tag{1.8}$$

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 an higher payoff than the defect strategy (*confess*), which

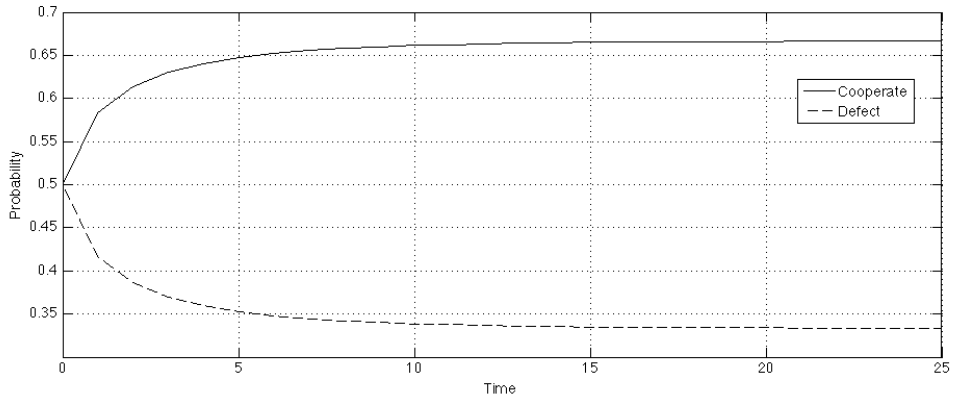


Figure 1.1: The dynamics of the repeated prisoner's dilemma.

is the equilibrium in the classical game theory framework. In fact, if the players play the game shown in Table 1.1 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:

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

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

where T is the transpose operator, required for P_2 which chooses its strategies according to the columns of the matrix in Table 1.1. 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 since they receive the same payoffs. Now we can compute the strategy space of a player at time $t+1$ according to equation (1.7):

x_1: $-1.25/-3 = 0.42$

x_2: $-1.75/-3 = 0.58$

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 1.1 we can see how the *cooperate strategy* increases over time, reaching a stationary point, which corresponds to the equilibrium of the game.

1.3 Population Games Dynamics

In the last section we have introduced some concepts of evolutionary game theory, defining how the strategic interactions occur repeatedly. These kind of games are

called *population games*, because they are defined on one or more populations of agents. In these games, each agent employs a procedure to decide when and how his strategies have to be changed. This procedure is called *revision protocol* and within a population game setting give rise to *evolutionary games dynamics* [39].

In population games we have a large number of interacting agents. In each interaction a game is played among two agents, in order to obtain a payoff. Each agent's payoff is determined by the co-player behavior and has little effect on the overall state of the system. Furthermore, each agent belongs to a population and each population has a finite set of possible strategies, which is identical for all the population members [39].

More formally, $\mathcal{P} = \{1, ..., p\}$ is a society composed of p populations and each population is composed by a mass of agents $m^p \geq 0$. The set of strategies of each populations, p is $S^p = \{1, ..., n\}$. The set of strategy distributions for each population p is $X^p = \{x^p \in \mathbf{R}_+^{n^p} : \sum_{i \in S^p} x_i^p = m^p\}$. The scalar x_i^p denotes the share of population agents choosing strategy $i \in S^p$ and X^p is the simplex in $\mathbf{R}^{n_p}$.

The payoff of a single strategy, $i \in S^p$ is denoted as $F_i^p = X \to \mathbf{R}$ and $F_i^p = X \to \mathbf{R}^{n^p}$ denotes the payoff for all strategies in S^p. The average payoff obtained by members of population p at social state x is,

$$\bar{F}^p(x) = \frac{1}{m^p} \sum_{i \in S^p} x_i^p F_i^p(x), \tag{1.9}$$

and the aggregated payoff of the entire society is defined as,

$$\bar{F}(x) = \sum_{p \in \mathcal{P}} \sum_{i \in S^p} x_i^p F_i^p(x) = \sum_{p \in \mathcal{P}} m^p \bar{F}^p \tag{1.10}$$

The best responses,

$$b^p(x) = \arg\max_{i \in S^p, ..., c} F_i^p(x), \tag{1.11}$$

are calculated in mixed strategies as follows,

$$B^p(x) = \{y^p \in \Delta^p : y_i^p > 0 \Rightarrow i \in b^p(x)\} \tag{1.12}$$

where Δ^p denotes the simplex in $\mathbf{R}^{n^p}$. $B^p(x)$ is the set of probability distributions whose supports contain only pure strategies that are optimal at x.

A social state can be defined as a Nash equilibrium if each agent in every population chooses a best response to x.

Theorem 2. *Every population game admits at least one Nash equilibrium.*

Proof. See [39]. □

In single population games we have the same situation described in Section 1.2, that is, if the payoff matrix of the game is A, then its population game is described by the linear map $F(x) = Ax$.

In two population, we have two strategy sets, $S^1 = 1, ..., n^1$, $S^2 = 1, ..., n^2$ and two payoff matrices, $A^1 \in \mathbf{R}^{n^1 \times n^2}$ and $A^2 \in \mathbf{R}^{n^1 \times n^2}$. In this case, every member of a population is matched to play the game (A^1, A^2) and the payoff function associated to each population is, $F^1(x) = A^1 x^2$ and $F^2(x) = (A^2)' x^1$. The entire game can be described as,

$$F(x) = \begin{pmatrix} F^1(x) \\ F^2(x) \end{pmatrix} = \begin{pmatrix} 0 & A^1 \\ (A^1)' & 0 \end{pmatrix} \begin{pmatrix} x^1 \\ x^2 \end{pmatrix} = \begin{pmatrix} A^1 x^2 \\ (A^2)' x^1 \end{pmatrix} \tag{1.13}$$

The previous example can be generalized to the case of games in p populations. In this context, the set of pure strategy profiles is defined as, $S = \prod_{q \in p} S^q$ and the payoff matrices of each population as $A = (A^1, ..., A^p)$. The payoff function of this game is,

$$F^p_{S^p}(x) = \sum_{s^{\neg p} \in S^{\neg p}} A^p(s^1, ..., s^p) \prod_{r \neq p} x^r_{s^r}, \tag{1.14}$$

where $S^{\neg p} = \prod_{q \neq p} S^q$. This implies that the behaviors of the players in the same population are not influenced reciprocally.

In this thesis, a multi-population framework was used to implement the word sense disambiguation task, in Chapter 2 and the document clustering games in Chapter 3, whereas the single population framework is employed by the dominant clustering algorithm, which was used in Chapter 3.

1.4 Evolutionary Dynamics

In general, a model is termed evolutionary if Evolutionary dynamics are based on three fundamental concepts: selection, replication, and mutation, The mechanism of selection is a force that favors some traits of behaviors rather than others. In game theory it is governed by payoffs, in a way in which, players that have obtained higher payoffs, that is, have determined traits or employ determined behavior, are selected preferentially from the population, for reproduction. The mechanism of reproduction is linked to the concept of replication. In fact, it increments the share of population with the traits or behaviors of selected players. The generation of new diversity in the population is accomplished by the mutation mechanism, which is a process that tends to slightly differentiate the replicated players.

These mechanisms have been implemented by the scientific community with the support of differential equations, which are able to model the evolution of a population over time. Two principles, which are common in these representations, are those of *inertia* and *myopia*. The first refers to the fact that players adjust their strategy sporadically (not each time they play a single game). The second refers to

the fact that agents revisit their strategies, only considering the actual state of the game, that is, their current payoff and their strategy distribution.

The mechanism which describes when and how the players update their behaviors, is based on the notion of *revision protocol* and follows the two assumptions of *inertia* and *myopia*, the agents wait a certain amount of time before they consider to update their decisions and this decision is based only on the current social state.

If we consider a population game, $F : X \to \mathbf{R}^{n^p}$, as defined in the previous section, a *revision protocol* ρ^p is a map $\rho^p : \mathbf{R}^{n^p} \times X^p \to \mathbf{R}_+^{n^p \times n^p}$. The scalar $\rho_{ij}^p(\pi^p, x^p)$ is called the conditional switch rate from strategy $i \in S^p$ to strategy $j \in S^p$, given payoff vector π^p and population state x^p.

The *revision protocol* of the replicator dynamics, introduce in Section 1.2, has the form,

$$\rho_{ij}^p(\pi^p, x^p) = \hat{x}_j^p r_{ij}^p(\pi^p, x^p), \tag{1.15}$$

where $\hat{x}_j^p$ is the share of population p playing strategy $j \in S$. More generally, we have that under the replicator dynamics, the percentage growth rate of each strategy $i \in S^p$ is equivalent to its excess payoff, $\hat{F}_i^p(x) = F_i^p(x) - \bar{F}^p(x)$, where $\bar{F}^p(x)$ is the average payoff.

2

Word Sense Disambiguation Games

2.1 Introduction

Word Sense Disambiguation (WSD) is the task to identify the intended meaning of a word based on the context in which it appears [40]. It has been studied since the beginnings of Natural Language Processing (NLP) [41] and also today it is a central topic of this discipline. This because it is important for many NLP tasks like text understanding [42], text entailment [43], machine translation [44], opinion mining [45], sentiment analysis [46] and information extraction [47]. 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 [48].

To solve this problem the research community has proposed several algorithms during the years, based on supervised [49, 50], semi-supervised [51, 52] and unsupervised [53, 54] learning models. Nowadays, although supervised methods perform better in general domains, unsupervised and semi-supervised models are gaining attention from the research community with performance close to the state of the art of supervised systems [55]. In particular knowledge-based and graph based algorithms are emerging as interesting approaches to resolve the problem [56, 57]. 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 consists in 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 [58]. 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, leading to the resolution of the problem [59, 60].

Our approach falls into 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 among our approach and state-of-the-art graph based approaches [61, 62, 57, 63] 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 which 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 [64, 65]. The interactions among units give rise to the emergence of properties which in our case, by assumption, 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 which share a syntactic or 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 [66].

This is illustrated in the sentences below:

- 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 which 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 *cricked* 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 syntactical or proximity information.

Our approach imposes that the senses of the words which 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 [67], within a Markow random field framework. They have shown that it is beneficial in combinatorial optimization problems. In our case, this imposition can preserve the textual coherence;

[1] A complete example of the disambiguation of the first sentence is given in Section 2.4.2

a characteristic which 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 simultaneously, using an underlying structure of interconnected links which models the interdependence between the words. In this way, 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 decided to use a powerful tool, derived from evolutionary game theory: the non-cooperative games (see Section 1.1). In our system, the nodes of the graph are interpreted as players, in the game theoretic sense (see Section 1.1), which 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 which 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 [35]. From our point of view, increasing the utility of a word means increasing the textual coherence, in a distributional semantics perspective [68]. In fact, it has been shown that collocated words tends to have a determined meaning [69, 70].

Game theoretic frameworks have been used in different ways to study the language use [71, 72] and evolution [73], 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 [74, 75], 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 definition, 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 combinatorial 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* [76]. In fact, as shown in the above two sentences, concentrating the disambiguation task of a word on highly collocated words, taking into account syntactic and proximity information allows the meaning interpretation to be guided towards senses which are strongly related to the word to be disambiguated and not to other words in its context.

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 a previous work we have used a similar algorithm in a semi-supervised scenario [77], casting the WSD task 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 [78], including syntactic and proximity information.

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 [77], is a bootstrapping graph based method which propagates the information from labeled words to unlabeled. In this article, we also provide a new semi-supervised version of the approach which can exploit the evidence from labeled words in corpora or the most frequent sense heuristic and does not require labeled words.

We tested our approach on different datasets in order to find the similarity measures which perform better and evaluated it against unsupervised, semi-supervised and supervised state-of-the-art systems. The results of this evaluation shows that our method performs well and can be considered as a valid alternative to current models.

2.2 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 interests, based on the relations that these words' senses have in the network [79].

A popular supervised WSD system, which has shown good performance in different WSD tasks, is *It Makes Sense* (IMS) [49]. 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 [80], DSO [81] and collected automatically exploiting parallel corpora [82]. 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 because, supervised systems require training data, which are difficult to obtain. In fact, producing labeled 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 [83], the performances of a supervised algorithm degrade substantially with the increasing 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 problems to adapt

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

to different contexts, because it depends on prior knowledge which makes the algorithm rigid, therefore cannot efficiently adapt to domain specific cases, when other optimal solution may be available [83].

One of the most common heuristics which allows to exploit labeled data such as SemCor [80] 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 can not handle senses with a low distribution which could be found in specific domains.

With these observations in mind Koeling et al. [84] 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 [54] and demonstrated that their approach outperforms the most frequent sense heuristic derived from general domain and labeled data.

This heuristic, for the unsupervised acquisition of the predominant sense of a word, consists in 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 labeled 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, instead of 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 of traditional algorithms which exploit the pairwise semantic similarity among a target word and the words in its context [85, 86, 87]. 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 combination.

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 employed both for the organization of the contextual information and for the computation of the relations among word senses. It allows to extract the structural properties of a text. Examples of this kind of approaches construct a graph from all the senses of the words in a text and then use connectivity measures in order to identify the most relevant word senses in the graph [57, 59]. Navigli and Lapata [59] 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 and if it exists, it adds all the nodes and edges on this path to the graph. The connectivity measures aim at finding the most important nodes in 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 measure and in particular, degree centrality and PageRank [88] (which is a variant of the eigenvector centrality measure) achieve the best results.

PageRank [88] is one of the most popular algorithm for WSD, in fact, it was implemented in different ways by the research community [89, 90, 61, 91]. 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 [88]. Exploiting this idea the ranking of the nodes in the graph can be computed iteratively with equation (2.1):

$$Pr = cMPr + (1 - c)v \tag{2.1}$$

where M is the transition matrix of the graph, v is a $N \times 1$ vector representing a probability distribution and c is the so called damping factor which 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.

An algorithm which tries to improve centrality algorithms is SUDOKU, introduced by Minion and Sainudiin [92]. 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 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 [66]. It approaches the WSD problem as a maximum a posteriori query on a Markov random field [93]. 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 dependency structure of the sentence, the frequency of the senses and the similarity among them.

A new graph based, semi-supervised approach, introduced to deal with multilingual WSD [94] and entity linking problems, is Babelfy [95]. 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 [96] to perform this task. Entity linking consists in disambiguating the named entities in a text and in finding the appropriate resource in an ontology which correspond to the specific entity, mentioned in a text. In this task, information from ontology such as those available in the Linked Open Data [97] are exploited to find the appropriate description of a named entity. Babelfy creates the semantic signature of each word to be disambiguated, that consists in collecting, from a semantic network, all the nodes related to a particular concepts, exploiting the global structure of the network. This process leads to the construction of a graph-based representation of the whole text. Then, it applies Random Walk with Restart [98] to find the most important nodes in the network, solving the WSD problem.

Approaches which are more similar to ours in the formulation of the problem have been described by Araujo [99]. The authors reviewed the literature devoted to the application of different evolutionary algorithm 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 [100]. It uses genetic algorithms [101] and memetic algorithms [102] in order to improve the performance of a *gloss-based* method. It is assumed 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 address the disambiguation problem in terms of space search is GETALP [103], it use 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 it is shown 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.

2.3 Word Sense Disambiguation as a Consistent Labeling Problem

WSD can be seen as a sense labeling task [40] 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 [104] in which the labeling process has to satisfy some constraints in order to be consistent. This approach gives 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 incongruences in the assignment of senses to related words.

As an illustrative example we can consider a binary CSP 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 solved if there exists 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 is the cardinality of domains and q is the cardinality of variables. Each element of the matrix $R_{ij}(\lambda, \lambda') = 1$ indicates the compatibility of the assignment $v_i = \lambda$ with the assignment $v_j = \lambda'$. R is the core part of the system. It is used to impose the constraints, in a way in which each label assignment is consistent.

The binary case described above assumes that the constraints are completely violated or completely respected. This is restrictive; in fact, in many real-word cases, it is more appropriate to have a weight which expresses the level of confidence about a particular assignment [74]. This consistency notion has been shown to be related to the Nash equilibrium concept in game theory [105]. 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. The computation of Nash equilibria has been introduce in Section 1.2.

2.4 WSD games

In this section we describe how the WSD games are formulated. We assume that each player $i \in I$ which 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 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 1.1. The games are played among two similar words, i and j, imposing only pairwise interaction among them. The payoff matrix Z_{ij} of a single game is defined as a sense similarity matrix, among the senses of word i and word j. The payoff function for each word is additively separable and is computed as described in Section 1.2.

Formulating the problem in this way we can apply the replicator dynamics equation (see Section 1.2, equation 3.16), 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 to this state not only considering its own strategies but also the strategies which its co-players are playing. When the system converges, for each player $i \in I$ is chosen the strategy with the highest probability (see equation below).

$$\phi_i = \arg\max_{h=1,...,c} x_{ih} \tag{2.2}$$

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 or he is not connected with other nodes. The latter assumption is not admitted in our framework. With equation 2.2 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 senses in it have the same probability.

2.4.1 Implementation of the WSD games

In order to run our algorithm we need the network which 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 which 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 which share a syntactical or proximity relation,
- extract, from I, the list C of all the possible senses, which represents 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 games,
- 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 section. In Section 2.4.1.1 we describe the graph construction procedure which we employed in order to model the geometry of the data. In Section 2.4.1.2 we explain how we implement the strategy space of the game, which allows each player to choose over a predetermined number of strategies. In Section 2.4.1.3 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 2.4.1.4 we describe the system dynamics.

2.4.1.1 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 j-th word and N is the total number of words retrieved. From I we construct a $N \times N$ 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 an useful tool for graph-based algorithms since it is treatable as 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 weights 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.

$$w_{ij} = sim(i_i, i_j) \; \forall i, j \in I : i \neq j \tag{2.3}$$

This choice is motivated by the fact that collocated words tend to have determined meanings [69, 70], and also because the computation of these similarities can be obtained easily. In fact, it only required a corpus in order to compute the a vast range of similarity measures. Furthermore, large corpora such as the BNC corpus [25] and the Google Web 1T corpus [24] 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, due to different text segmentation techniques or spelling differences. In this case we use query expansion techniques in order to find an appropriate substitute [106]. Specifically, we use WordNet to find alternative lexicalizations of a lemma, choosing the one which co-occurs more frequently with the words in its context.

The information obtained from a similarity measure can be enriched taking into account the proximity of the words in the text and the syntactic structure of the sentence. The first task can be achieved augmenting the similarities among a target word and the k words that appear on its right and on its left, where k is a parameter that with small values can capture fixed expressions and with large values can detect semantic concepts [107]. The second task can be achieved using a dependency parser to obtain the syntactical relations among the words in the target sentence.

Essentially, a dependency parser takes as input a sentence and gives as output a dependency structure. The structure is represented by a directed labeled graph with the main verb of the sentence as its root. The edges of the graph connect words which have a dependency relation and its labels specify the relation type [108]. The dependency relation is asymmetric and connects a syntactically subordinate word, called dependent with another word on which it depends, called head word. The directed edge goes from the head word to the dependent, making the relation asymmetric.

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 while the use of a distributional approach allows us to exploit the relations of semantically correlated words. This is particularly useful when the dependency structure is poor; for example when it connects words to auxiliary or modal verbs.

2.4.1.2 Strategy space implementation

The strategy space of the game is created using a knowledge base to collect the sense inventories $M_i = 1, \ldots, m$ of each word in a text, where m is the number of senses associated with word i. Then is created the list $C = 1, \ldots, c$ 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:

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

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 1.1. The initialization of each mixed strategy space can either be uniform or take into account information from sense labeled corpora.

2.4.1.3 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 chose that sense, and play the strategy associated with it.

The $c \times c$ sense similarity matrix Z is defined in equation (2.4).

$$z_{ij} = ssim(s_i, s_j) \; \forall i, j \in C : i \neq j \tag{2.4}$$

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 among 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.

2.4.1.4 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 (3.16). In each iteration of the system each player plays a game with its neighbors N_i according to the similarity graph W and the payoffs are calculated as follows:

$$u_i(e^h, x) = \sum_{j \in N_i} (w_{ij} Z_{ij} x_j)_h \tag{2.5}$$

and

$$u_i(x) = \sum_{j \in N_i} x_i^T (w_{ij} Z_{ij} x_j) \tag{2.6}$$

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.

2.4.2 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, tag and parse the sentence using the Stanford parser [109]; then we extract the content words which have an entry in WordNet 3.0 [110], constructing the list of words to be disambiguated: {is, financial, institution, river, bank}. Once we identified the target words we computed the pairwise similarity for each target word. For this task we used the Google Web 1T 5-Gram Database [24] to compute the modified Dice coefficient[3] [111]. With the information derived by this process we can construct a similarity graph (Figure 2.1(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 (Figure 2.1(b)) or the proximity information derived by a simple n-gram model (Figure 2.1(c)).

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

In Figure 2.1(d) it is represented the final form of the graph for our target sentence, in which we have combined the information from the similarity graph and from the n-gram graph. The weights in the similarity graph are increased by the mean weight of the graph if a corresponding edge exists in the n-gram graph and not include a stop-word[4]. As we can see in Figure 2.1(b) and 2.1(c), in both graphs there is an edge between the words *bank* and *river* meaning that this relation is more important that the others.

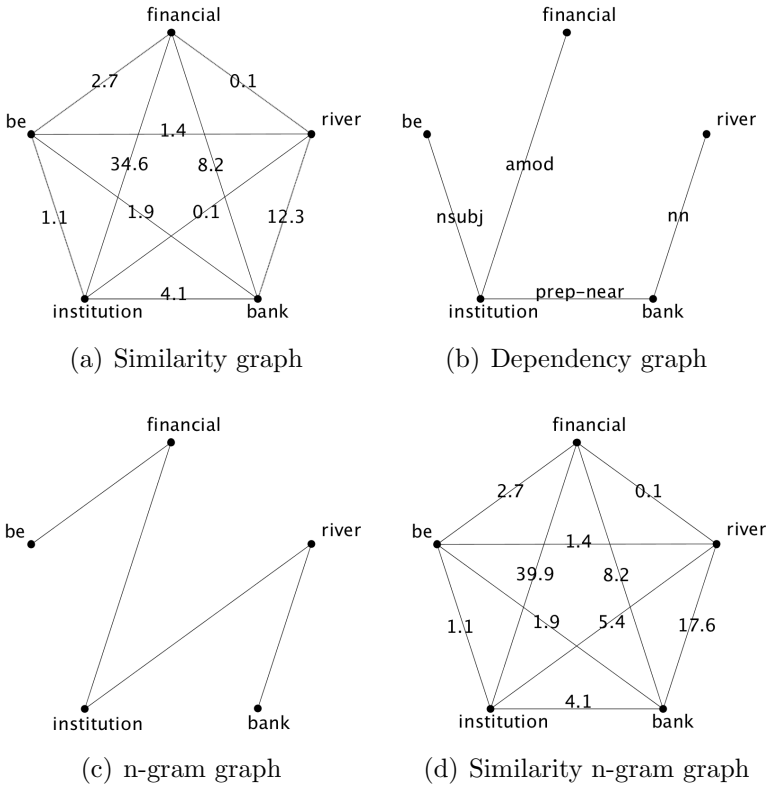
After the pairwise similarities between the words are computed we access a lexical database in order to get the sense inventories of each word so that each word can be associated with a predefined number of senses. For this task, we use WordNet 3.0 [110]. Then for each unique sense in all the sense inventories we compute the pairwise semantic similarity, in order to identify the affinity among all the pairwise sense combination. This task can be done using a semantic similarity or relatedness measure [5]. For this example, we used a variant of the *gloss vector measure* [112], the *tf-idf*, described in Section 2.5.1.2.

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

[3]Specifically we used the service provided by the Corpus Linguistics group at FAU Erlangen-Nürnberg, with a collocation span of 4 words on the left and on the right and a collocates with minimum frequency: 100.

[4]A more accurate representation of the data could have been implemented using the dependency graph instead of the n-gram graph or both of them; 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 an n-gram model can implicitly detect syntactical relations

[5]Semantic similarity and relatedness measures are discussed in Section 2.5.1.1 and 2.5.1.2



(a) Similarity graph (b) Dependency graph

(c) n-gram graph (d) Similarity n-gram graph

Figure 2.1: Four graph representations for the sentence: there is a financial institution near the river bank. (a) a similarity graph constructed using the modified Dice coefficient as similarity measure over the the Google Web 1T 5-Gram Database [24] to weight the edges. (b) graph representation of the dependency structure of the target words using the Stanford dependency parser [109]. (c) 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 of size one word. (d) a weighted graph which combine the information of the similarity graph and the n-gram graph. The edges of similarity graph are augmented by its mean weight if a corresponding edge exists in the n-gram graph and not include a stop-word.

any prior information about the senses distributions. Now the system dynamics can be started. In each iteration of the dynamics each player play a game with its neighbors obtaining a payoff for each of its strategies according to equation (2.5) and 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 (3.16).

We present the dynamics of the system created for the example sentence in Figure 2.2. 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

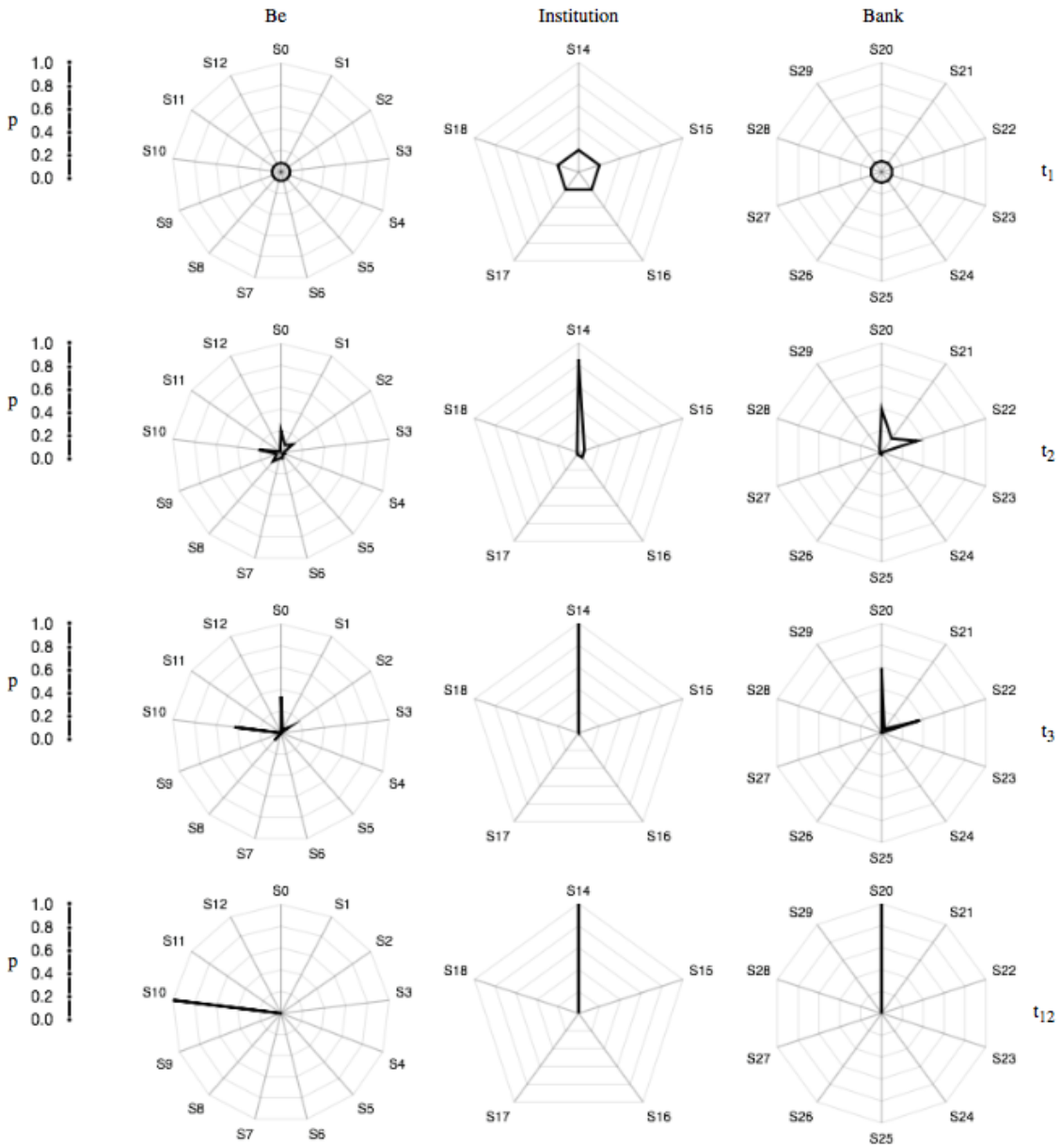


Figure 2.2: System dynamics for the words: *be*, *institution* and *bank* at time step 1,2,3 and 12 (system convergence). The strategy space of each word is represented as a regular polygon of radius 1, where the distance from the center to any vertex represents the probability associated with a particular 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.

each word are equiprobable, but as soon as the games are played some sense starts 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 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* which are not compatible with the senses of the other words in the text and therefore their values decrease rapidly.

The most interesting phenomenon which 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 [110], and can be used in different context and domains, to indicate, among the other things, a financial institution (s_{22} in Figure 2.2) or a sloping land (s_{20} in Figure 2.2). 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 2.2 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 chose 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.

2.5 Experimental Evaluation

In this Section we describe how the presented method has been tested and compared with state-of-the-art systems[6]. In Section 2.5.1 we describe the datasets which we have used for the evaluation of our model and the settings which we have used to test it. In Section 2.5.2 the results of our experiments with an unsupervised setting are presented and in Section 2.5.3 the results using a semi-supervised setting are presented. In Section 2.5.4 the detailed results of our experiments on each dataset are presented; finally, in Section 2.5.5 we compare our results with state-of-the-art systems. The results are provided as F1, computed according to equation 2.7.

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \cdot 100 \tag{2.7}$$

[6]It is possible to download the code of the algorithms and the datasets used in this chapter at http://www.dsi.unive.it/~tripodi/wsd

F1 is a measure which 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.

2.5.1 Evaluation Setup

We evaluated our algorithm with four general domain datasets: Senseval-2 fine grained english all-words (SE2) [113], Senseval-3 fine grained english all-words (SE3) [114], SemEval-2007 fine grained all-words (SE07FG) [115], and SemEval-2007 coarse grained english all-words (SE07CG) [116][7].Furthermore we evaluated our approach on two Entity Linking datasets, SemEval-2013 task 12 (S13) [117] and KORE50 [118][8], using as knowledge base BabeNet.

In the next sections, we describe the similarity measure which we used to test our approach, introduce WordNet [110] and BabelNet [96], presenting the semantic and relatedness measure calculated on these resources. Finally, we explain how we initialize the strategy space for the WSD games.

2.5.1.1 Distributional similarity measures

We evaluated our algorithm with different similarity measures in order to find the measure which performs better, the results of this evaluation are presented in Section 2.5.2. Specifically for our experiments we used eight different measures: the *Dice coefficient* (*dice*) [119], the *modified Dice coefficient* (*mDice*) [111], the pointwise mutual information (*pmi*) [120], the *t-score* measure (*t-score*) [120], the *z-score* measure (*z-score*) [121], the *odds ration* (*odds-r*) [122], the *chi-squared* test (*chi-s*) [123] and the *chi-squared* correct (*chi-s-c*) [124].

The measures which we used are presented in Figure 2.4 where the notation refers to the standard contingency tables [125] used to display the observed and expected frequency distribution of the variables, respectively on the left and on the right of Figure 2.3.

2.5.1.2 Semantic measures

We used WordNet [110] and BabelNet [96] as knowledge bases to collect the sense inventories of each word to be disambiguated.

[7]SE2 has been downloaded from www.hipposmond.com/senseval2, SE3 from http://www.senseval.org/senseval3, SE07FG from http://nlp.cs.swarthmore.edu/semeval/tasks/ and SE07CG from http://lcl.uniroma1.it/coarse-grained-aw

[8]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/

	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 2.3: Contingency tables of observer frequency (on the left) and expected frequency (on the right).

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

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

$$chi\text{-}s = \sum_{ij} \frac{(O_{ij}-E_{ij})^2}{E_{ij}} \qquad chi\text{-}s\text{-}c = \frac{N(|O_{11}O_{22}-O_{12}O_{21}|-N/2)^2}{R_1R_2C_1C2}$$

Figure 2.4: Association measures used to weight the similarity graph W.

2.5.1.2.1 Semantic measures calculated with WordNet WordNet [110] 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 which 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 which have an opposite meaning; meronymy connects the concept which is part of a given concept with it; and holonymy relates a concept with its constituents. Furthermore, each synset is associated to a definition and gives the morphological relations of the word forms related to it. Given the popularity of the resource many parallel project have been developed. One of them is *eXtended WordNet* [126] 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 in account a wider range of relations such as the *is-a-part-of* or *is-the-opposite-of*.

The similarity measure which we used are the *wup similarity* [127] and the *jcn measure* [128]. These measure are based on the structural organization of WordNet and compute the similarity among two senses s_i, s_j according to the depth of the two sense in the lexical database and that of the most specific ancestor node, *msa*, of the two senses. The *wup similarity*, described in equation (2.8), 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 (2.9), where IC is the information content of a concept c derived from a corpus[9] and computed as $IC(c) = log^{-1}P(c)$.

$$ssim_{wup}(s_i, s_j) = 2 * depth(msa)/(depth(s_i) + depth(s_j)) \tag{2.8}$$

$$ssim_{jcn}(s_i, s_j) = IC(s_1) + IC(s_2) - 2IC(msa) \tag{2.9}$$

The semantic relatedness measures, which 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 occur in the gloss and n is the total number of different words (*types*) in the corpus[10]. This representation allows to project each vector into a *vector space* where it is possible to conduct different kind 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 (2.10), 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: $\sqrt{\sum_{i=1}^{n} w_i^2}$.

$$\cos\theta \frac{v_i \cdot v_j}{||v_i||||v_j||} \tag{2.10}$$

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 an high similarity we transform this distance measure into a similarity measure with $1 - cos(v_i, v_j)$.

The procedure to compute the semantic similarity among two synsets has been introduced by Patwardhan and Pedersen [112] as *Gloss Vector measure* and we used

[9] We used the IC files computed on SemCor [80] 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 dataset.

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

it with six different variations for our experiments. The six variations are named: tf-idf, tf-idf$_{\text{ext}}$, tf-idf$_3$, vec, vec$_{\text{ext}}$, and vec$_3$. The difference among them relies on the way the gloss vectors are constructed. Since the synset gloss is usually short we used the concept of *super-gloss* as in [112] 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 [129]. 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*). It uses the same WordNet version of the dataset and only the relations in WordNet. *tf-idf$_{ext}$* uses the same information of *tf-idf* plus the relations derived from eXtended WordNet [126]. *tf-idf$_3$* is equivalent to *tf-idf* but uses a different knowledge base: WordNet 3.0. *vec* employs the same WordNet version of the dataset on which it is used and the computation of the co-occurrence is computed with a standard BoW approach. *vec$_{ext}$* uses the same information of *vec* plus the relations from eXtended WordNet. *vec$_3$* are the same of *vec*, but the synsets are mapped on WordNet version 3.0.

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 occur. 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 occur 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.

2.5.1.2.2 Semantic measure calculated with BabelNet and NASARI BabelNet [96] 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 to link *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[11] [130], 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*

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

in WordNet and Wikipedia and then constructs two vectors to construct 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. The descriptions of the related concepts are used to construct a word representation of b, then it is used lexical specificity[12] [131] in order to extract the most representative words to construct the first vector and the most representative synsets to construct the second vector.

In this article, we computed the similarity among 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 which best represent e 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 case of named entity disambiguation, since it includes many entities which are not included in WordNet. To the other hand, it is difficult to use it in all-words sense disambiguation tasks, since it includes only WordNet synsets which 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, which is common to find in all-words sense disambiguation tasks.

2.5.1.3 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, which 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 Z we can obtain the partial semantic similarity matrix for each pair of player, $Z_{ij} = m \times n$, where m and n are the senses of i and j in Z.

In a previous work [78] we did not use this information, instead we used labeled data points to propagate the class membership information over the graph. In this new version the use of the semantic information made the algorithm completely unsupervised.

2.5.1.4 Strategy space implementation

Once the pairwise similarities between the words and their senses, stored in the two matrices W and Z, are obtained, we can begin to describe the strategy space of each player. It can be initialized with equation (2.11) which follows the constraints described in Section 1.2 and assigns to each sense an equal probability.

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

$$s_{ij} = \begin{cases} |M_i|^{-1}, & \text{if sense } j \text{ is in } M_i. \\ 0, & \text{otherwise.} \end{cases} \tag{2.11}$$

This initialization is used in the case of unsupervised WSD since 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 with its rank, concentrating a higher probability on senses which have a high frequency. To model this kind of scenario we used a geometric distribution which gives us a decreasing probability distribution. In equation (2.12) we defined this new initialization as follows,

$$s_{ij} = \begin{cases} p(1-p)^{r_j}, & \text{if sense } j \text{ is in } M_i. \\ 0, & \text{otherwise.} \end{cases} \tag{2.12}$$

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 (2.12) 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) [132] 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 which can be queried with a lemma and a part of speech to obtain the list of possible sysnets associated with the specified lemma and a part of speech. The returned synsets are listed in decreasing order of frequency and can be used as ranking system by our algorithm.

2.5.2 Experiments with an unsupervised setting

Tables 2.1, 2.2, 2.3, 2.4 show the results for the datasets SE07, SE07FG, S3 and S2 respectively. In these experiments, we used the distributional and semantic similarities described in Section 2.5.1. We recall that the distributional similarities are computed over the target words using the BNC corpus [25], to weight the similarity graph W and that the semantic similarities are computed over the senses, using a different version of WordNet, to calculate the payoff matrix Z, in order to measure the strength of compatibility between the senses of two words which are involved in a game.

From these series of experiments we can see how different combinations of measures affect the results. The aim of the experiment is to discover what is the combination of measures which performs better. From Table 2.5 we can see that on average the combination of measures with the best performance is *mdice* and $tfidf$. This combination has the higher arithmetic mean and in two datasets (SE07 and

	dice	mdice	pmi	t-score	z-score	odds-r	chi-s	chi-s-c
$tfidf$	80.2	**80.3**	75.0	77.7	78.3	79.2	73.1	79.5
$tfidf_{ext}$	77.7	77.4	73.3	73.3	75.9	77.3	71.5	77.4
$tfidf_3$	79.5	79.3	74.3	77.7	78.2	78.7	72.7	79.2
vec	80.2	**80.3**	74.5	77.4	79.0	79.1	73.0	79.2
vec_{ext}	75.0	75.5	72.2	72.3	75.1	75.5	70.1	75.7
vec_3	79.1	78.7	73.9	75.7	77.8	78.3	71.4	78.4
wup	60.9	60.2	55.1	62.4	62.8	61.4	61.0	62.0
jcn	39.5	38.9	35.8	40.9	42.0	42.9	39.4	42.9

Table 2.1: Results as F1 for SE07.

S3) it gives the best results. In S2 the best distributional similarity measures are *odds-r* and *hi-s-c*. In SE07FG the semantic measure which performs better is the *vec* whereas the distributional similarity with highest values is *z-score*.

It is important to note that the best results are given by semantic measures computed over the WordNet glosses. In particular, the best measures are those computed extracting the glosses from the same WordNet version of the dataset. However, when the synsets are mapped to WordNet 3.0 the semantic measures achieve lower results. Probably due to some errors in the mapping schema which we adopted for the computation of the measures $tf\text{-}idf_3$ and vec_{ext}, as reported by Agirre et al. [61]. It has been observed that, in none of the tested datasets these measures have performed better than the measures computed on the same WordNet version of the dataset. The same behavior can be observed for the measures which employed *eXtended WordNet* [126] ($tf\text{-}idf_{ext}$ and vec_3), since it is required to map each synset to WordNet 2.0 in order to use this resource.

The semantic measures with the worst performance are *wup* and *jcn*. The reason why these measures are not suited for our model is that they can be computed only on synsets which have the same part of speech. This limitation has affected the results of our system because without a payoff matrix the games played between two words with different parts of speech have no effect on the dynamics of the system, since their values are zeros. This affects the performance of our system in terms of recall, since many words in the dataset can not be calculated, they tend to remain on the central point of the simplex.

The distributional measure with the best performance is *mdice*. On average, it performs better than any other semantic measures in association with all the semantic measure which we have used. The measure with the worst performance is *pmi*. This can be explained given the fact that *pmi* tends to takes high values when one word in the collocation has low frequency. This behavior does not imply high dependency and therefore can compromise the results of the games.

	dice	mdice	pmi	t-score	z-score	odds-r	chi-s	chi-s-c
$tfidf$	42.9	43.3	38.1	44.2	45.3	43.3	41.3	43.7
$tfidf_{ext}$	42.9	42.9	37.4	43.1	44.6	43.3	40.7	45.1
$tfidf_3$	42.4	42.0	38.6	44.2	45.3	43.3	42.2	44.4
vec	46.4	46.2	39.9	45.3	**46.8**	46.2	42.6	46.2
vec_{ext}	45.3	45.7	37.3	42.0	44.2	43.7	41.5	44.4
vec_3	46.6	45.9	40.5	45.1	46.4	45.5	43.5	46.2
wup	34.0	34.2	28.4	33.7	34.1	32.8	36.9	33.0
jcn	16.7	15.7	18.0	16.7	16.2	17.1	17.1	17.7

Table 2.2: Results as F1 for SE07FG.

	dice	mdice	pmi	t-score	z-score	odds-r	chi-s	chi-s-c
$tfidf$	58.5	**59.1**	49.8	56.7	56.7	57.9	55.3	58.3
$tfidf_{ext}$	56.1	56.6	48.1	53.2	55.1	55.0	51.7	55.7
$tfidf_3$	58.0	58.1	50.7	55.8	56.3	57.6	55.0	57.6
vec	57.5	57.6	51.9	53.2	55.5	56.7	53.3	56.3
vec_{ext}	52.3	52.6	46.9	48.0	50.2	51.8	48.6	52.3
vec_3	57.2	56.5	50.8	53.1	54.9	56.5	52.7	56.5
wup	44.6	44.9	38.6	45.2	45.4	45.3	43.6	45.4
jcn	26.7	27.1	27.3	27.2	26.7	26.3	28.2	27.0

Table 2.3: Results as F1 for SE3.

	dice	mdice	pmi	t-score	z-score	odds-r	chi-s	chi-s-c
$tfidf$	60.2	61.2	57.8	61.0	61.1	**61.9**	60.3	**61.9**
$tfidf_{ext}$	57.4	57.1	55.2	56.2	57.1	57.2	56.4	57.4
$tfidf_3$	59.3	59.3	57.4	60.1	59.7	59.8	59.5	59.9
vec	58.4	59.4	56.7	59.2	58.9	59.0	59.5	58.9
vec_{ext}	54.6	55.1	51.8	52.7	54.3	54.5	53.5	54.6
vec_3	57.6	57.7	55.7	57.1	57.6	57.4	57.3	57.4
wup	43.5	42.9	42.8	44.9	43.3	43.2	43.7	43.2
jcn	33.0	33.2	35.5	34.6	32.7	33.1	34.2	34.1

Table 2.4: Results as F1 for SE2.

	dice	mdice	pmi	t-score	z-score	odds-r	chi-s	chi-s-c
$tfidf$	60.5	**61.0**	55.2	59.9	60.4	60.6	57.5	60.9
$tfidf_{ext}$	58.5	58.5	53.5	56.5	58.2	58.2	55.1	58.9
$tfidf_3$	59.8	59.7	55.3	59.4	59.9	59.8	57.4	60.3
vec	60.6	60.9	55.8	58.8	60.1	60.2	57.1	60.1
vec_{ext}	56.8	57.2	52.0	53.8	56.0	56.4	53.4	56.7
vec_3	60.1	59.7	55.2	57.8	59.2	59.4	56.2	59.6
wup	45.8	45.6	41.2	46.6	46.4	45.7	46.3	45.9
jcn	29.0	28.7	29.1	29.9	29.4	29.9	29.7	30.4

Table 2.5: Average results as F1 for SE07, SE07FG, SE3, SE2.

2.5.3 Experiments with a semi-supervised setting

Once discovered the combination of measures with best results, we conducted a different series of experiments in order to test the performance of our system with a semi-supervised setting. As we introduced in Section 2.4.1.2 and 2.5.1.4 the strategy space of our system can be initialized with a uniform distribution in unsupervised setting or can use the information about the frequencies of the synsets derived from sense labeled corpora, in semi-supervised setting. In the latter setting the dynamics of the system can be initialized assigning a higher probability to senses which appear more frequently in a corpus and a lower probability to unfrequent senses in semi-supervised setting.

The first experiment with this new approach aims at discovering the best value for the parameter p which determines the decrease of the probabilities associated with the ranked senses. We have done this experiment keeping fixed the combination of similarity measures which we have discovered with our first series of experiments and changing the values of p, in the interval $[0.05, 0.95]$. Figure 2.5 shows that the performance for the fine grained datasets (SE07FG, S3 and S2) increases with values of p between 0.05 and 0.5. With higher values of p the results tend to regularize or to decrease slightly. The best results for the fine grained dataset reaches values of p between 0.4 and 0.5. For the coarse grained dataset we can observe a different behavior. In fact, the results in this case tend to decrease with increasing values of p. This phenomenon can be explained by the fact that the coarse grained dataset has a different structure which maps each word to a cluster of senses and not to a single sense as in the fine grained case. For this reason concentrating the probabilities according to the rank of the single senses is not effective in this case. Furthermore, for this dataset our system is able to obtain results which are higher than the results which can be obtained using the most frequent sense heuristic; and also for this reason, the use of information derived from sense labeled corpora does help to increase the performance of the system; to the contrary, it makes the results worse.

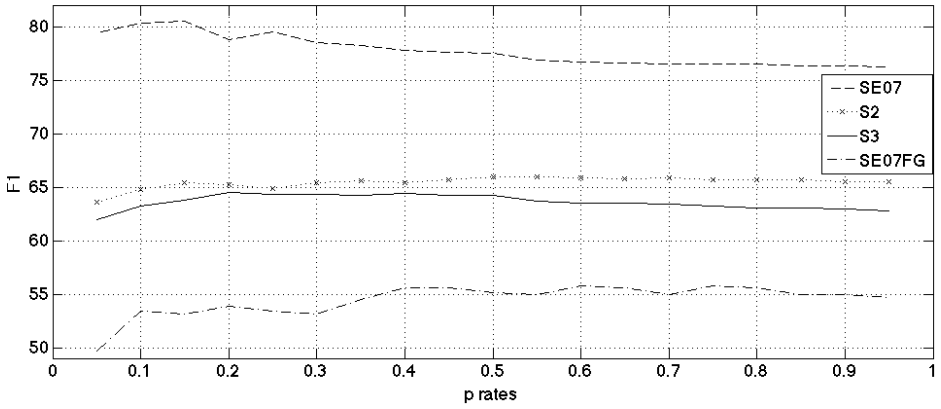


Figure 2.5: Results as F1 on SE07, SE07FG, S3 and S2 with changing values of the p parameter.

Employing a semi-supervised setting is beneficial for the fine grained dataset. The results for these datasets are always higher than the results obtained with the unsupervised setting, whatever is the value taken by the p parameter. From Figure 2.5 we can also see that the variance in the results of the three fine grained datasets is low. To the contrary, the results of the coarse grained dataset are always lower than the results obtained without the use of prior knowledge. The experiments for this dataset show that only with low values of p it is possible to obtain good performance. In fact, as we can see with values of p greater than 0.25 the results tend to decrease. This behavior confirms that the use of prior knowledge is not effective when the performance of the unsupervised system is well above the performance of the most frequent sense heuristic, as it is shown in the next section.

2.5.4 Detailed results

Table 2.6 shows the detailed results as F1 for the four datasets which we used for the experiments. The table includes the results for the two implementation 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) [132]. As we can see the best performance of our system are obtained on nouns, on all the datasets. This is in line with the state-of-the-art systems because in general the nouns have lower polysemy and higher inter-annotator agreement [113]. Furthermore, our method is particularly suited for nouns. In fact, the disambiguation of nouns benefits from a wide context and local collocations [133].

We obtained low results on verbs, on all datasets. This, as pointed out by Dang [134], is a common problem not only for supervised and unsupervised WSD systems

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

As we have anticipated in the previous section, the use of prior knowledge is beneficial in case of fine grained WSD. As we can see in Table 2.6 using a semi-supervised setting improves the results of 5% on S2 and S3 and of 12% on SE07FG. The big improvement obtained on SE07FG 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 sense labeled dataset is beneficial. To the contrary, the results obtained for SE07 with a semi-supervised setting are below those of the unsupervised systems and this because the structure of the datasets is different and also because the results obtained with the unsupervised setting are well above the most frequent sense.Finally, these series of experiments show that our system performs better than the most common sense heuristic on all datasets and this with the support of prior knowledge.

2.5.5 Comparison to state-of-the-art algorithms

Table 2.7 shows the results of our system and the results obtained by state-of-the-art systems on the same datasets. We compared our method with supervised, unsupervised and semi-supervised system on four datasets. The supervised systems are *It makes sense* [49] (*Zhong10*), an open source WSD system based on support vector machines [135]; and the best system which participated to each competition (*Best*). The semi-supervised systems are: *IRST-DDD-00* [136], 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 [66]; *Nav05* [52] a knowledge based method which exploits manually disambiguated word senses to enrich the knowledge base relations; PPR_{w2w} [61] a random walk method which 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 which 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 which does not use sense tagged resources.

The results shows that our unsupervised system performs better than any other unsupervised algorithm in all datasets. In SE07 and SE07FG the difference is minimal compared with PPR_{w2w} and *Nav10*, respectively; in *S3* and *S2* the difference is more substantial compared to both unsupervised systems. Furthermore, the performance of our system is more stable on the four datasets, showing a constant improvement on the state-of-the-art.

The comparison with semi supervised systems shows that our system performs always better than the most frequent sense heuristic when we use information from sense labeled corpora. We can note a strange behavior on SE07, when we use prior

SemEval 2007 coarse grained - SE07					
Method	All	N	V	A	R
WSD^{uns}_{games}	80.3	85.5	71.2	81.5	76.0
WSD^{ssup}_{games}	77.6	77.9	71.2	83.1	85.1
MFS	76.3	76.0	70.1	82.0	86.0

SemEval 2007 fine grained - SE07FG					
Method	All	N	V	A	R
WSD^{uns}_{games}	43.3	49.7	39.9	-	-
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
WSD^{ssup}_{games}	64.6	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	67.7	41.0	62.8	64.8
WSD^{ssup}_{games}	65.8	72.3	43.3	72.0	74.8
MFS	65.6	72.1	42.4	71.6	76.1

Table 2.6: Detailed results as F1 for the four datasets 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 employing the most frequent sense heuristic (MFS). Detailed information about the performance of the systems on different part of speech are provided: nouns (N), verbs (V), adjectives (A), adverbs (R).

	SE07	SE07 (N)	SE07FG	S3	S2
$Nav10^{uns}$			43.1	52.9	
PPR^{uns}_{w2w}	80.1	83.6	41.7	57.9	59.7
WSD^{uns}_{games}	**80.3**	**85.4**	**43.3**	**59.1**	**61.2**
$IRST\text{-}DDD\text{-}00^{ssup}$				58.3	
MFS^{ssup}	76.3	77.4	54.7	62.8	65.6
$MRF\text{-}LP^{ssup}$			50.6	58.6	60.5
$Nav05^{ssup}$	**83.2**	84.1		60.4	
PPR^{ssup}_{w2w}	81.4	82.1	48.6	63.0	62.6
WSD^{ssup}_{games}	77.6	77.9	**56.5**	**64.6**	**65.8**
$Best^{sup}$	82.5	82.3	**59.1**	65.2	**68.6**
$Zhong10^{sup}$	82.6		58.3	**67.6**	68.2

Table 2.7: Comparison with state-of-the-art algorithms: unsupervised (*uns*), semisupervised (*ssup*) and supervised (*sup*). *MFS* refers to the MFS heuristic computed on SemCor on each dataset and *BEST* refers to the best supervised system for each competition. The results are provided as F1.

knowledge the performance of our semi-supervised system are lower than our unsupervised system and state-of-the-art. This is because on this dataset the performance of our unsupervised system are better than the results than can be achieved by using labeled data to initialize the strategy space of the semi supervised system. On the other three datasets we can note a substantial improvement in the performance of our system, with stable results higher than state-of-the-art systems.

Finally we can note that the results of our semi supervised system, on the fine grained datasets, are close to the performance of state-of-the-art supervised systems, with values in average below of 2.8%. We can also note that the performance of our unsupervised system on the nouns of the SE07 dataset are higher than the results of the supervised systems.

2.5.6 Experiments with BabelNet

For the experiments on the Entity Linking datasets we used BabelNet to collect the sense inventories of each word to be disambiguated, the $m - dice$ 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 are computed using the cosine similarity measure, described in Section 2.5.1.2. The only differences with the experiments presented in Section 2.5.2 are 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 5 languages (we used only English). All the nouns in these texts are annotated using BabelNet, with a total number of 1931 entities to be disambiguated (English dataset). KORE50 consists of 50 short English sentences with a total number of 144 mentions manually annotated using YAGO2 [137]. 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 dataset contains highly ambiguous mentions, which is 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.

The results of these experiments are shown in Table 2.8, where it is possible to see that the performances of our system are 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 an high level of ambiguity, such as those proposed by KORE50. To the contrary, PPR_{w2w} performs poorly on the same dataset. This because, as attested in [95], it disambiguates the words independently, without imposing any consistency requirements.

The good performances of our approach are also due to the good semantic representations provided by NASARI, in fact, it is able to exploit a richer source of information, Wikipedia, which provides a larger coverage and a wider source of

	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	-

Table 2.8: Comparison with state-of-the-art algorithms on Entity Linking. The results are provided as F1 for S13 and as accuracy for KORE50.

information than WordNet alone.

2.6 Conclusions

In this chapter we have introduced a new method for WSD based on game theory. We have provided an extensive introduction on the WSD task and explained the motivations behind the choice to model, the WSD problem as a constraint satisfaction problem. We have conducted an extensive series of experiments to find out the combination of similarity measures which performs better in our framework. We have also evaluated our system with two different implementation and compared our results with state-of-the-art systems, for WSD and Entity Linking.

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 exploited also 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 to always find the best labeling assignment.

Our system in addition to have a solid mathematical and linguistic foundation, has demonstrated to perform well compared with state-of-the-art system 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 to 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 impose to respect the sense compatibility among each sense before to assign 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 weighting the context ensures to respect the syntactic structure of a sentence and to disambiguate each word according to the syntactical context in which it appears. This is because the meaning of a word in a sentence does not depend on *all* the words contained in the sentence but only on those which share a syntactical relation and those with which enjoy a high distributional similarity.

3

Document Clustering Games

3.1 Introduction

This chapter is about *document clustering*, which is a subset of the larger field of *cluster analysis* (or *clustering*). *Cluster analysis* is an unsupervised learning task which consists in partitioning a set of objects into groups called clusters [138]. In document clustering the objects, which have to be grouped, are textual documents, such as: books, web pages, tweets, news, emails, etc.. This operation is aimed at organizing documents automatically. It is particularly useful in this period, characterized by a pervasive use of information communications technologies. In fact, the use of social networks and the internet, by a large audience, is increasing the amount of data to handle. It helps to develop browsing mechanisms and knowledge management systems, trying to organize things in a way similar to that used by humans, taking into account similarities and differences between objects and organizing them accordingly.

From a cognitive perspective, clustering can be considered as one of the earliest abilities developed by human beings [139]. In fact, it is required by more specific abilities, such as, the construction of basic units of knowledge and the development of social skills [140]. All these activities require the ability to discover similarities and differences among objects or patterns, in order to mentally organize and categorize the reality.

Clustering can be related to *categorization*, the task of organizing objects into classes. It starts with the recognition of similar features and organizes similar objects in the same class. The main difference among these two tasks is that when the categorization is performed, the number of categories, in which the objects to be categorized have to be placed, is given and also some correlations between objects and classes are provided, whereas in clustering this information has to be discovered.

Categorization has a long history, it was first introduced in the context of western philosophy by Plato, who defined it as the task of grouping objects with similar properties. Aristotle systematized this idea proposing an enumeration of the most general kinds (categories) into which the objects of the world can be divided. The aristotelian categorization system can be interpreted as a taxonomy of the reality. In this framework, the categories are characterized by a set of properties which have

to be completely owned by an object in order to be inserted into a particular class. Another important characteristic of this categorization system is that the classes are mutually exclusive.

A more recent approach to this task, called *prototype theory*, has been proposed by Eleanor Rosch [141]. Within this interpretation, the objects are grouped together according to *prototypes*. The difference with the classical approach is that in this framework an object falls into a category if it has a certain number of properties. Another important difference is that in this framework concepts (categories) have a probabilistic structure which describes its relevant properties (*conceptual cores*).

The theoretical formulation of the classification problem, proposed by Rosh [142, 141], is in line with modern approaches to clustering, in which each object is represented as a set of features. This feature set determines the structure of the clusters and each object is inserted in the cluster which has a feature structure similar to it.

The aim of clustering is the same as categorization and relies on the discovery of natural groupings. The only difference is that in clustering the categorical structure of the objects to be clustered must be induced and depends on the features of the objects to be clustered. For this reason, an approach which is able to discover non trivial structures in the data, able to discriminate among many different grouping, is required.

In this chapter we evaluate different game theoretic methods to perform document clustering. We show what the advantages are of using a game theoretic framework for document clustering, highlighting the differences with state-of-the-art methods. We will describe the problems encountered and will propose new solutions to deal with these problems.

3.2 Clustering

Clustering can be defined as an unsupervised classification of patterns [143]. It consists in the recognition of data items which by means of some similarity criterion can be grouped together and can be considered separated by the rest of the data. Each group is classified with a different label, which identifies a specific cluster (category) of objects. A cluster can be described as a maximally coherent set of data items. The data items must satisfy an internal criterion which imposes that all elements belonging to a cluster must be highly similar to each other and an external criterion which imposes that all items belonging to different clusters must be highly dissimilar [138].

The earliest application of clustering methods was introduced in the field of cultural anthropology, with studies aimed at discovering affinities among different cultures and at organizing them accordingly [144]. These studies investigate the relationship among tribes, comparing the presence or the absence of some traits in different tribes.

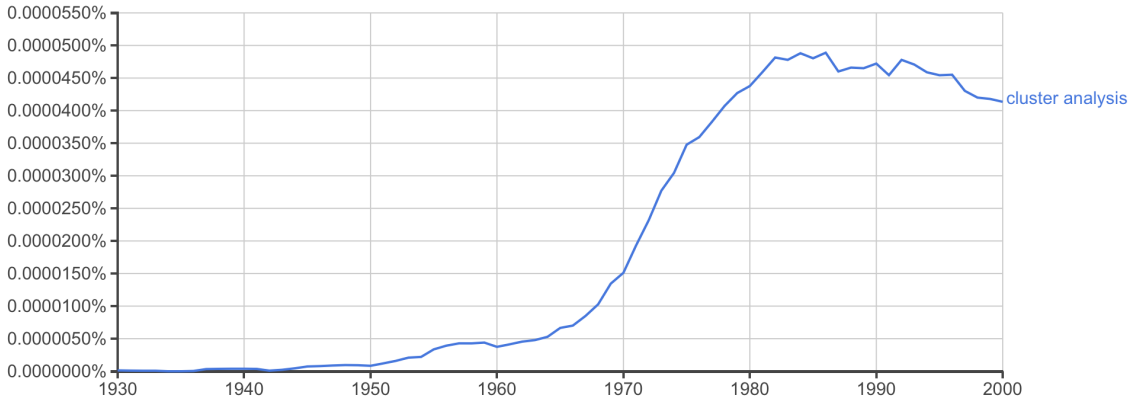


Figure 3.1: The vertical axis indicates the frequency of the two expressions: *clusteranalysis*, in the English scientific literature, as collected by Google. The horizontal axis indicates the year of the measurements.

The term *cluster analysis* was introduced by Robert Tryon, a behavioral psychologist, in 1939 [145]. He is famous for his studies on hereditary trait inheritance, conducted through laboratory intelligence tests on rats. He introduced a mathematical procedure for organizing objects, based on similarity measurements stored in a correlation matrix, an archetypal form of clustering, performed computing manually the factor analysis of the correlation matrix. He introduced a procedure in which the observations are first collected into two groups and then these two groups are partitioned again and again until each observation forms a group by itself. This approach can be considered a progenitor of modern hierarchical clustering. During the same period, also in the psychology field, it has been published a seminal work on personality classification, with the help of clustering notions [146].

As shown in Figure 3.1, even if the concept of *clustering analysis* was introduced in the 1930's, it is only during the 1960's, that the studies on clustering started to become popular, attracting the attention of the scientific community. One of the most important works of this period was published by Sokal and Sneath, in 1963 [147]. In this study the authors show how it is possible to create a taxonomy using cluster analysis instead of using a subjective evaluation of the properties of the objects to be organized. This work emphasized the utility of automatic methods for the analysis of data and since this work, *clustering analysis* has been used in different fields, from biology to medicine, from engineering to economics, and social sciences.

3.2.1 The Clustering Model

Formally, given a set of input data $X = \{x_i, ..., x_n\}$, where $x_i = \{x_{i1}, ..., x_{im}\}^T \in \mathbb{R}^d$ where each element x_{i1} is a measure (feature or dimension) relative to pattern i, the aim of hard clustering models is to find a K-partition, $C = C_1, ..., C_K$ of X, such

that,

1. $C_i \neq \emptyset, i = 1, ..., K$;
2. $\cup_{i=1}^{K} C_i = X$;
3. $C_i \cap C_j = \emptyset, i, j = 1, ..., K$ and $i \neq j$;

Within this formulation each pattern is allowed to belong to one cluster. To the contrary, in *fuzzy clustering* it is produced an association matrix $W = W_{i,k} \in [0, 1], i = 1, ..., n, j = 1, ..., K$, whose elements $w_{i,k}$ indicate the degree to which each pattern i belongs to cluster k.

Clustering process is composed of several steps:

1. sample
2. feature extraction
3. pattern (dis)-similarity computation
4. clustering algorithm implementation/choice
5. cluster validation
6. results interpretation

The first step of the process regards the construction of the dataset to cluster, which consists in the sampling of a set $X = \{x_i, ..., x_n\}$ of n entities. From this sample (dataset), the *feature extraction* has to be conducted. It consists in the representation of the entities as a set of features. Different kinds of data have different measurement techniques and different data representation, for this task. When the measurements has been conducted, the *feature selection* is conducted. This process consists in selecting the features which better describe the dataset. Once the dataset is defined and the data are represented, it is necessary to select a function able to measure the *(dis)-similarity* among the patterns, also here, different kinds of data have different *(dis)-similarity*. Almost all clustering algorithms are based on similarity/proximity measures [148]. Also in this step, there are different measures for different kind of data. For example, for textual data the *cosine distance* is used, whereas for image data the *euclidean distance* is one of the most popular measure. Usually, in this step the data are represented as an $n \times n$ matrix, called proximity/similarity matrix, whose (i, j) elements represent the pairwise similarity or dissimilarity among pattern i and j.

Clustering algorithm implementation/choice depends on the problem at hand. In fact, it does not exists a general framework for clustering and many algorithms have been proposed during the years to solve different problems. *Cluster validation* refers to the analysis of the clustering produced by different algorithms or by the

same algorithm using different parameter values (*parameter tuning*). The last step regards *results interpretation*. During this step the results are validated against a benchmark. It is also possible to consult an expert in the field of the problem at hand to interpret the data partition.

3.2.2 Clustering Methods

Clustering algorithms can be classified into two different groups: hierarchical clustering and partitional clustering. This distinction takes into account the structure of the clusters which these two approaches produce. In hierarchical clustering the clusters produced form a tree-like structure, in which a cluster can be included in another. In partitional clustering the objects are divided into some pre-specified number of clusters without the hierarchical structure [148].

In this chapter we will focus on partitional clustering. One of the most popular algorithm for partitional clustering is *K-means*. This algorithm assigns each observation to the cluster with the nearest mean. In its basic formulation, it selects randomly the k means from the samples, then these points serve as prototypes for each cluster and the partitions are performed according to this initialization. After the partitioning is performed, the algorithm recalculates the cluster prototypes, according to the partition obtained in the previous step. The samples are then associated to new clusters, considering the new prototypes. These steps of selection and partitioning, are repeated until there is no change in the assignments.

The function which k-means tries to minimize is the following,

$$\arg\min_{C} \sum_{i=1}^{K} \sum_{x \in C_i} ||x - \mu_i||^2 \tag{3.1}$$

where μ_i is the mean of points in S_i. The k-means algorithm is very simple and efficient, one problem is that it is difficult to identify the initial partition. A choice which is very important because it leads the algorithm to very different final partitioning.

Another important group of clustering algorithms is characterized by approaches based on *graph theoretic* models. Within this framework, we have a graph $G = (V, E, w)$, where $V = \{1, ..., , n\}$ are the nodes of the graph, which represent the samples, $E = V \times V$ is the edge set of the graph, in which the graph connections are stored and $w : E \to \mathbb{R}^*_+$ is the (positive) weight function, which encodes the proximity/similarity information among nodes. In its basic representation, the graph G is reduced to its undirected and unweighted form, according to Equation 3.2.

$$D_{ij} = \begin{cases} 1, & \text{if } D(x_i, x_j) < d_0. \\ 0, & \text{otherwise.} \end{cases} \tag{3.2}$$

Where d_0 is a threshold value, which ensures that only relevant proximity measures are taken into account. In this case the problem of clustering is reduced to the prob-

lem of finding maximally connected subgraphs (components) or maximally complete subgraphs (cliques) [148]. Other approaches use weighted graphs to represent the data and exploit different graph-theoretic concepts to find good partitioning, such as minimum spanning tree [149] or minimum cut [150]. However, these models are not able to find maximal cliques with this representation, because this operation has been generalized only on unweighted graphs. The capacity to find maximal cliques is desirable in many cases, because it has been pointed out in [151, 152] that the concept of maximal clique is the strictest definition of a cluster. An algorithm which has generalized the notion of maximal cliques is the dominant set clustering algorithm [5]. Such an algorithm is used in this chapter and will be described in Section 3.4.

A different branch of research on clustering is based assumes that the data to be clustered are part of some unobserved probability distribution. The algorithms developed around this assumption are based on a *generative model* and are used to find a clustering that best agrees with the underlying model. In the field of document clustering there are several approaches, which are based on probabilist models. They differ in the distribution used to model the data and use an expectation–maximization algorithm to learn the parameter required to generate the distribution.

3.3 Document clustering

Document clustering is a particular kind of clustering which involves textual data. The objects to be clustered can have different characteristics, varying in length and content. The most popular applications of document clustering aims at organizing tweets [153], news [154], novels [155], medical documents [156] etc.. It is a fundamental task in text mining, with different applications which span from document organization to language modeling [157].

The most important challenge for this task lies in the sparsity and the high dimensionality of text data [158]. This is due to the *bag-of-words* (BoW) model, which is commonly used in NLP and Information Retrieval to represent texts. BoW is a model which represents each document, in a text collection as a vector, which is indexed according to the vocabulary of the corpus. The vocabulary of the corpus is represented as the set of unique words, which appear in a text collection. BoW does not take into account grammar or word ordering; it just records in each entry of the vectors the presence, the count or the frequency of a determined word.

The BoW representation can be described in matrix format. To develop such representation, is constructed a $D \times T$ matrix C, where D is the number of documents in the corpus and T the number of elements in the vocabulary of the corpus. This kind of representation is called *document-term matrix*, its rows are indexed by the documents and its columns by the vocabulary terms. Each cell of the matrix $tf(d,t)$, indicates the frequency of the term t in document d.

This representation can lead to a high dimensional space. In fact, as the Zipf's law indicates [159, 160], the vocabulary size increases as corpus size increases. An important consequence of the BoW model is the sparseness of the vectors representing the documents. In fact, if the vocabulary of the corpus is large or the texts are short the vectors will have many zero entries, compromising the representation of the data. Another important aspect which characterizes the BoW model consists in the fact that it does not incorporate semantic information. For this reason, synonyms are represented as different features. In this case, two documents, which use different words to express similar concepts, are represented differently. The lack of semantic information constitutes problems also with the homonyms. In fact, two documents which use homonyms words to express different concepts, will be treated as similar.

The sparseness and high number of features, obtained with the BoW approach, can result in bad representations of the data. This is essentially because it gives the same importance to all the features and also because it treats in the same manner documents of different dimension [158]. For these problems the scientific community has proposed different approaches, which tries to balance the importance of each feature and to reduce the dimensionality of the feature space.

The documents to be represented by the BoW model are typically preprocessed. The preprocess starts breaking the stream of each text into elements (usually words) called tokens. Each *token* is then converted into a *type*, which is the base form of a word, represented by its lemma. Lemmas constitute the elements of the vocabulary of the corpus and are used as features to describe the texts. Given the fact that many words in the language have no relevant semantic information (such as articles or pronouns), these words are not considered as features. It is common to have a list of these words, called *stopwords*, and to remove them when the texts are preprocessed.

Other heuristics can be adopted to remove words which are not good at describing text. For example, the words which appear in the majority of the documents can be removed, in fact, their contribution is not beneficial, they are not discriminative. Another technique which is largely used to smooth the importance of a feature in a text is the *term frequency - inverse document frequency* (tf-idf) weighting method [157]. The basic idea of this method is to give less importance to features which are shared by many documents. This method takes as input a document-term matrix C and update it with the following equation,

$$tf\text{-}idf(d,t) = tf(d,t) \cdot log\frac{D}{df(d,t)} \tag{3.3}$$

where $df(d,t)$ is the number of documents containing the term t. Then the vectors are normalized so that no bias can occur because of the length of the documents. Tf-idf can give a better representation of the texts but the problems relative to the lack of semantic information, the high dimensionality and the sparseness of the matrix remain.

A method to deal with these problems is called Latent Semantic Analysis (LSA) [161]. It is largely used in the Information Retrieval and NLP community and it is able to capture semantic information between terms which can be used to measure the similarity among two documents. It is also able to reduce the dimensionality of the data representation, concentrating its attentions only on the most important features.

The semantic information is obtained projecting the documents into a *semantic space*, where two documents can be considered similar even if they do not share any term, but have terms which are semantically related. The relatedness of two terms can be computed considering the context in which they appear. If two terms have a similar context, they are considered related, from a distributional semantics point of view [68].

The advantage in using LSA is that it does not require an external knowledge base to infer the correlation among terms. Instead, it uses the corpus itself to find patterns of co-occurrence. On the other side, it is difficult to use it in real application, since it is required to update the matrices each time new documents are added to the corpus.

The co-occurrence patterns and the dimensionality reduction are calculated by means of a Single Value Decomposition (SVD) of the term by documents matrix or tf-idf matrix. It constructs an approximation of the original matrix, preserving the similarity among the documents. This technique tries to decompose the term by document matrix D in:

$$D = U\Sigma V^T, \tag{3.4}$$

where Σ is a diagonal matrix with the same dimensions of D and U and V are two orthogonal matrices. The dimensions of the feature space is reduced to k, taking into account the first k of the matrices in Equation (3.4).

Dimension reduction is used to merge together terms that have a similar semantics (occur in a similar context), furthermore it is used, as a preliminary process, to identify and disambiguate terms with multiple meanings and to provide a lower-dimensional representation of documents that reflects concepts instead of raw terms [162].

LSA uses spectral decomposition to identify a lower-dimensional representation that maintains semantic properties of the documents. Once the new representation is obtained, it is possible to compare the documents, using different proximity/similarity measures, in a low-dimensional space. With this representation, it is possible to conduct the analysis of documents at a conceptual level, overcoming the drawbacks of term-based analysis [162]

Another method for dimension reduction, is topic modeling. It includes probabilistic latent semantic analysis (PLSA) [163] and latent Dirichlet allocation [164]. In these methods, to conduct dimension reduction, it is used a probabilistic model, which is able to find co-occurring terms that can be considered as semantic topics

in a collection of texts [162].

These models try to overcome the problems of BoW representation and are aimed at discovering the abstract topics underlying a collection of documents. By abstract topic it is intended a set of words which appear in a specific context and can be interpreted as the vocabulary used to describe a particular topic. In this view, documents are seen as mixtures of topics and topics as probability distributions over words. Essentially, they try to discover the probabilistic procedure by which documents can be generated. PLSA [163] models the probability of the co-occurrences of words and documents as a mixture of conditionally independent multinomial distributions,

$$P(w_i, d_j) = \sum_k p(w_i|z_k)p(z_k)p(d_j|z_k), \tag{3.5}$$

where the probabilities factors are normalized,

$$\sum_{i=1}^{m} p(w_i|z_k) = 1, \quad \sum_{j=1}^{n} p(d_j|z_k) = 1, \quad \sum_{k=1}^{K} p(z_k) = 1 \tag{3.6}$$

This model has been generalized by Blei et al. [164] adding a Dirichlet prior on the topic distribution of the documents. Topic models are useful because discovering the distributions of topics can be used directly to cluster documents. In fact, topics can be seen as sets of features which characterize a cluster.

Another important method which is largely used for dimension reduction and clustering is Nonnegative Matrix Factorization (NMF). It was introduced by Lee et al. [6], to learn the part of an image and the semantic features of a text. NMF differs from methods based on SVD, in that the semantic space derived by NMF does not need to be orthogonal. Furthermore, the reduced matrix, obtained with this method, takes only non-negative values [165].

In general, NMF takes as input a data matrix X, and produces two textitk-rank nonnegative matrices U_k and V_k, so that $U_k V_k^T$ provides an approximation to X. The aim of this approach is to find the two matrices U and V which minimize the following objective function:

$$J = \|X - UV^T\|^2 \tag{3.7}$$

where $\| \cdot \|$ denotes the matrix Frobenius norm.

The matrices, provided by NMF, indicate how terms are associated to topics (matrix V) and how documents are associated to topics (matrix U). For clustering purpose, U can be used to assign each document to a cluster (hard clustering), with the following method,

$$k_i = argmax(k_{i,1}, ..., k_{i,K}) \tag{3.8}$$

There is a strict correlation between PLSA and NMF, in fact, it has been demonstrated that the two approaches are equivalent and optimize the same objective function [166].

These methods identify the relationships among terms and the dimensions of a latent space. Topic models are usually interpreted by inspecting the term-topic associations, finding the vocabulary of a determined topic and associating each document to the topic with which shares the largest number of terms [162].

A popular graph-based algorithm for document clustering is CLUTO [167], which uses different criterion functions to partition the graph into a predefined number of clusters. The problem with partitional approaches is that it is necessary to give as input the number of clusters to extract. The underlying assumption behind models based on matrix factorization, such as Non-negative Matrix Factorization (NMF) [6, 166] is that words which occur together are associated with similar clusters. [166] demonstrated the equivalence between NMF and Probabilistic Latent Semantic Indexing, a popular technique for document clustering. A general problem, common to all the approaches described, involves the temporal dimension. In fact, for these approaches is difficult to deal with datasets which evolve over time and in many real world applications documents are streamed continuously.

With our approach we try to overcome this problem, simulating the presence of some clusters into a dataset and classifying new instances according to this information. We also try to deal with situations in which the number of clusters is not given as input to our algorithm. The problem of clustering new objects is defined as a game, in which we have labeled players (clustered objects), which always play the strategy associated to their cluster and unlabeled players which try to learn their strategy according to the strategy that their co-players are choosing. In this way the geometry of the data is modeled as a similarity graph, whose nodes are documents (players), and the games are played only between similar players.

3.4 Dominant Set Clustering

As introduced in Section 3.2.2, dominant set clustering generalizes the notion of maximal clique from unweighted undirected to edge-weighted graph. Essentially, this generalization is relevant because it enables to extraction of compact structures from a graph in an efficient way. Furthermore, it has no parameters and can be used on symmetric and asymmetric similarity graphs. It offers measures of clusters cohesiveness and measures of vertex participation to a cluster. It is able to model the definition of a cluster, which states that a cluster should have high internal homogeneity and that there should be high inhomogeneity between the samples in the cluster and those outside. [168].

To model these notions we can use a graph G, with no self loop, represented by its corresponding weighted adjacency matrix $A = (a_{ij})$ and consider a cluster as a subset of vertices in it, $C \subseteq V$. The average weighted degree of node $i \in C$ with

regard to C is defined as,

$$awdeg_C(i) = \frac{1}{|C|}\sum_{j\in C} a_{ij} \tag{3.9}$$

We can also define the average similarity among a vertex $i \in C$ and a vertex $j \notin C$ as,

$$\phi(i,j) = a_{ij} - awdeg_C(i) \tag{3.10}$$

The weight of node i with respect to C can be defined as,

$$W_C(i) = \begin{cases} 1, & \text{if } |C| = 1 \\ \sum_{j\in C\setminus\{i\}} \phi_{C\setminus\{i\}}(j,i) W_{C\setminus\{i\}}(j), & \text{otherwise} \end{cases} \tag{3.11}$$

and the total degree of C is,

$$W(C) = \sum_{i\in C} W_C(i). \tag{3.12}$$

This measure gives us the relative similarity among vertex i and the vertices in $C\setminus\{i\}$, with respect to the overall similarity between the vertices in cluster $C\setminus\{i\}$. $W_C(i)$ gives us the measure of vertex participation to a cluster, which should be homogeneous for all $i \in C$. More formally, the conditions which enable the dominant set to realize the notion of cluster described above are:

1. $W_C(i) > 0$, for all $i \in C$
2. $W_{C\cup\{i\}}(i) < 0$, for all $i \notin C$

the first refers to the internal homogeneity of the cluster and the second refers to the external inhomogeneity.

A way to extract structures from graphs, which reflects the two conditions described above, is given by the following quadratic form:

$$f(x) = x^T A x \tag{3.13}$$

Within this interpretation, the clustering task is interpreted as that of finding a vector x, that maximize f. The vector x is is a probability vector, whose components express the participation of nodes in the cluster, so we have the following program:

$$\begin{array}{c} \text{maximize } f(x) \\ \text{subject to } x \in \Delta \end{array} \tag{3.14}$$

where,

$$\Delta = \left\{ x \in \mathbb{R}^n : x \geq 0 \text{ and } \sum_{i=1}^{n} x_i = 1 \right\} \tag{3.15}$$

A (local) solution of program 3.14 corresponds to a maximally cohesive cluster [168]. Furthermore we have,

Theorem 3. *If S is a dominant subset of vertices, then its weighted characteristic vector x^S is a strict local solution of program 3.14.*

Proof. See [5]. □

By formulating the problem in this way, the solution of program 3.14 can be found using the replicator dynamic equation, which we have introduce in Section 1.2 and 2.4.1.4, and propose again here, in the context of dominant set clustering,

$$x_i(t+1) = x_i(t)\frac{(Ax)_i}{x(t)^T Ax(t)} \tag{3.16}$$

In the dominant set framework, the clusters are extracted sequentially from the graph and a peel-off strategy is used to remove the data points belonging to an determined cluster, until there are no points to cluster or a certain number of clusters have been extracted.

3.5 Document Clustering Games

This section describes how document clustering games are formulated. The steps undertaken to resolve the task are as follows: data preparation, graph construction, clustering, strategy space implementation, clustering games and results evaluation. They are described in separate paragraphs below.

3.5.1 Data preparation

Each document i in a corpus D is represented with a BoW approach. From this data representation it is possible to adopt different dimension reductions techniques, such as LSA and NMF (see Section 3.3), to achieve a more compact representation of the data. The new vectors will be used to compute the pairwise similarity among documents and to construct, with this information, the proximity matrix W. As measure for this task, it was used the cosine distance (equation (2.10)), introduced in Section 2.5.1.2.

3.5.2 Graph construction

The proximity matrix obtained, in the previous step, can be used to represent the corpus D as a graph G, whose nodes are the documents in D and whose edges are weighted according to the similarity information stored in W. Since, the cosine distance acts as a linear kernel, considering only similarity between vectors under the same dimension, it is common to use a kernel function to smooth the data and

transform the proximity matrix W into an affinity matrix S [169]. This operation is also useful because it allows to transform a set of complex and nonlinearly separable patterns into patterns linearly separable [170]. For this task we used the classical Gaussian kernel,

$$\hat{s}(i,j) = exp\left\{-\frac{s_{ij}^2}{\sigma^2}\right\} \tag{3.17}$$

where s_{ij} is the dissimilarity among pattern i and j computed with the cosine distance and σ is a is a positive real number which determines the kernel width, and affects the decreasing rate of $\hat{s}$. This parameter is calculated experimentally, since the nature of the data and the clustering separability indices of the clusters is not known [171]. The clustering process can also be helped using graph Laplacian techniques. In fact, these techniques are able to decrease the weights of the edges between different groups of nodes. We use the normalized graph Laplacian, in some of our experiments, which is computed as follows:

$$L = D^{-1/2}\hat{S}D^{-1/2} \tag{3.18}$$

where D is the degree matrix of $\hat{S}$. Once we have matrix L we can reduce the number of nodes in it, so that document games are played only among high similar nodes, this refinement is aimed at modeling the local neighborhood relationships among nodes and can be done with two different methods, the ϵ-neighborhood graph, which maintains only the edges which have a value higher than a predetermined threshold, ϵ; and the k-nearest neighbor graphs, which orders the edges weights in decreasing order and maintains only the first k.

The effect of these processes is shown in Figure 3.2. On the main diagonal of the matrix it is possible to recognize some blocks which represent the clusters of the dataset. The values of those points is low in the cosine matrix, since it encodes the proximity of the points. Then the matrix is transformed into a similarity matrix by the Gaussian kernel, in fact, the points on the main diagonal in this representation are high. In the Laplacian matrix, it is possible to note that some noise has been removed from the matrix, the elements far from the diagonal appear now clearer and the blocks near the diagonal now are more uniform. Finally the k-nn matrix remove many nodes from the representation, giving a clear picture of the clusters.

We used the Laplacian matrix for the experiments with the dominant set, since this framework requires that the similarity values among the elements of a cluster are very close to each other. The k-nn graph has been used to run the clustering games, since this framework does not need many data to classify the points of the graph.

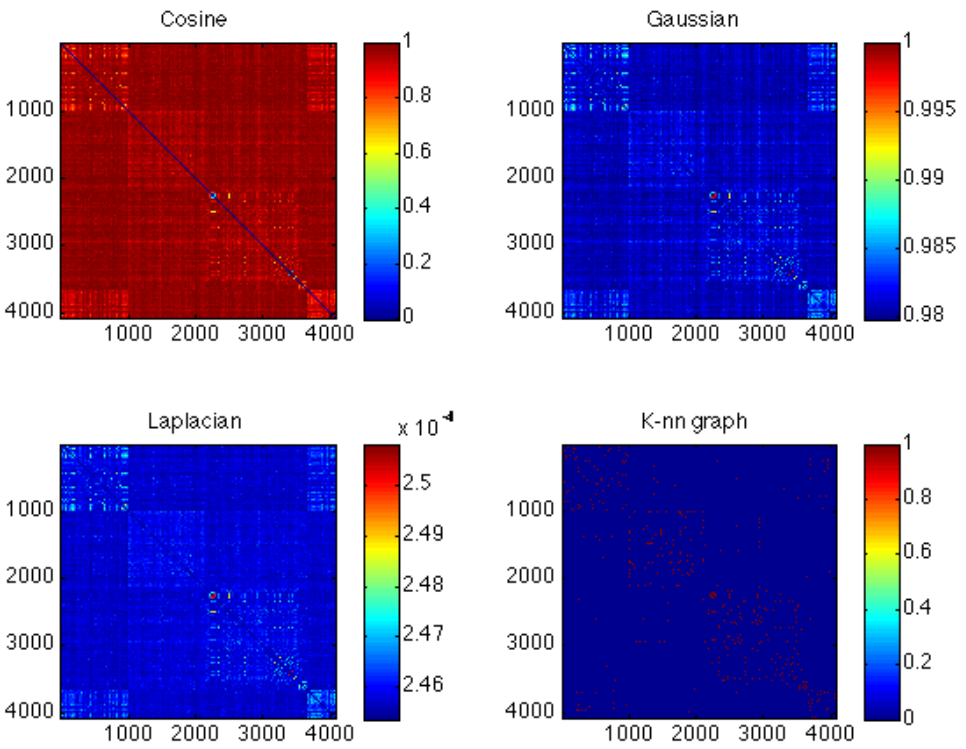


Figure 3.2: Different data representations for a dataset with 5 classes of different size.

3.5.3 Clustering

The clustering phase has been conducted using two different methods, the dominant set and the NMF. With the dominant set we have developed different implementations, giving as input the number of clusters to extract and also without this information, which is not common in many clustering approaches. Different dimension reductions have also been tested on this task, to evaluate the best representation of the data, for the problem at hand.

As anticipated in Section 3.3, the NMF algorithm, aims at reducing the number of features to characterize the objects in a dataset. With this approach, it is common to reduce the number of features to the number of clusters to extract [166, 165]. This process leads to the definition of a matrix $M = nk$, where n is the number of objects and k is the number of clusters. The rows of this matrix represent the objects in the dataset and the columns indicates the degree to which a particular feature is relevant for a particular document.

3.5.4 Strategy space implementation

In the previous step it has been shown that with the proposed approach, the dominant set clustering does not cluster all the nodes in a graph and that the clustered points are used to supply information to the others nodes. Within this formulation, it is possible to adopt different strategies to try to cluster the remaining nodes. On the other hand, with NMF there is not a hard partition of the data, because the documents are associated to topics with some degree of membership. In both cases evolutionary dynamics can be employed to cluster the unlabeled points or to refined the clustering obtained.

In the case of dominant set the strategy space of each player can be initialized as follows,

$$s_{ij} = \begin{cases} K^{-1}, & \text{if node } i \text{ is unlabeled.} \\ 1, & \text{if node } i \text{ has label } j, \end{cases} \tag{3.19}$$

where K is the number of clusters to extract and K^{-1} ensures that the constraints, required by a game theoretic framework (see Section 1.2), are met.

Regarding NMF, as shown in Section 3.3, it produces two matrices, U and V, the first associates documents to clusters, the second words to clusters. We are interested in U, since it is used to cluster the documents (see equation 3.8). We can initialize the strategy space of the NMF games, using this matrix, since it indicates the strength of association among a document and a clusters. In order to constrain each vector to lie on the standard simplex, we use the following equation to initialize the strategy space of the NMF games,

$$s_{ij} = \frac{u_{ij}}{\sum_{j=1}^{K} u_{ij}} \tag{3.20}$$

These initializations are aimed at improving the performances of dominant set clustering and NMF, starting the dynamics of the games not on the center of the K-dimensional simplex, Δ_K , as it is customary to do (see equation 2.5.1.4), but on a different interior point, which does not compromise that the dynamics will arrive to a Nash equilibrium (see Theorem 1).

3.5.5 Clustering games

Once the graph, which models the pairwise similarity among the players, and the strategy space of the games, has been created, it is possible to describe more in detail how the games are formulated.

It is assumed that each player $i \in I$, which participates in the games is a document in the corpus and that each strategy, $S_i = 1, ..., K$ is a particular cluster. The players can choose a determined strategy among the set of strategies, each expressing a certain hypothesis about its membership in a cluster and K being the total number of clusters available. We consider S_i as the mixed strategy for player i as described in Section 1.1. The games are played among two similar documents, i and j, imposing only pairwise interaction among them. The payoff matrix Z_{ij} of the games, contrary to what has been done in the word sense disambiguation games (see Section 2.4), is defined as a an identity matrix of rank K. This choice is motivated by the fact that, here all the players have the same strategy space, we do not know in advance, what is the range of classes to which the players can be associated, (excluding the labeled points obtained in the clustering phase. For this reason we have to assume that a document can belong to all classes.

In this setting the best choice for two similar players is to be clustered in the same class, which is expressed by the entry $Z_{ij} = 1, i = j$, of the identity matrix. In these kinds of games, called *imitation games*, the players try to learn their strategy by osmosis, learning by their co-players. Within this formulation, the payoff function for each player is additively separable and is computed as described in Section 1.1. Specifically, in the case of clustering games, there are labeled and unlabeled players, which can be divided in two disjoint sets, I_l and I_u, denoting labeled and unlabeled players, respectively. These groups can be divided further, considering the strategy that labeled players play without hesitation. In formal terms, there are K disjoint subsets, $I_l = \{I_{l|1}, ..., I_{l|K}\}$, each subset denoting the players that always play their kth pure strategy.

The labeled players always play the strategy associated to their cluster, because their strategy lays on a corner of the simplex, which corresponds to a rest point [172]. in this setting, labeled players do not play the games to maximize their payoffs, they have already a determined strategy. Only unlabeled players play the games, they have to decide their cluster membership (strategy) and they exploit the information provided by labeled players, in fact, they act as bias over the choices of unlabeled players. We recall that the games, formulated in these terms, always have a Nash equilibrium in mixed strategies [2] and that the adaptation of the players to the

proposed strategic environment is a natural consequence in game dynamics, given the fact that each player gradually adjusts his choices according to what other players do [39]. Once the equilibrium is reached, the cluster of each player i, corresponds to the strategy s_{ij}, with the highest probability (see equation 2.2).

The payoffs of the games are calculated as usual with equations 1.5 and 1.6, which in this case, with labeled and unlabeled players, are defined as,

$$u_i(e_i^k) = \sum_{j \in I_u} (L_{ij} A_{ij} x_j)_h + \sum_{k=1}^{K} \sum_{j \in I_{l|k}} L_{ij} A_{ij}(h,k) \tag{3.21}$$

and,

$$u_i(x) = \sum_{j \in I_u} x_i^T L_{ij} A_{ij} x_j + \sum_{k=1}^{K} \sum_{j \in I_{l|k}} x_i^T (L_{ij} A_{ij})_k. \tag{3.22}$$

In the case of NMF, each player has a different strategy space distribution. It is possible to have players, which have a definite strategy and act, as well as, labeled players in the case of clustering games with dominant set. There are also players with a uniform strategy distribution and players with a slightly preference for some strategy. In the latter case these players can influence other players from the beginning and can be influenced by other players, leading to the consolidation of their initial strategy or to the modification of it, caused by the strategic environment.

3.6 Experimental Setup

In this section we describe the experiments conducted with dominant set clustering and with NMF. We measured the performances of the systems using the accuracy measure and the normalized mutual information (NMI).

The accuracy is calculated with the following equation,

$$AC = \frac{\sum_{i=1}^{n} \delta(\alpha_i, map(l_i))}{n} \tag{3.23}$$

where n denotes the total number of documents in the test, $\delta(x,y)$ equals to 1, if x and y are clustered in the same class; $map(L_i)$ maps each cluster label l_i to the equivalent label in the benchmark. The best mapping is computed using the Kuhn-Munkres algorithm [173].

The NMI measure was introduced by Strehl and Ghosh [174] and indicates the level of agreement between the clustering C provided by the ground truth and the clustering C' produced by a clustering algorithm. The mutual information (MI) between the two clusterings is computed with the following equation,

$$MI(C,C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \cdot log_2 \frac{p(c_i, c'_j)}{p(c_i) \cdot p(c'_j)} \tag{3.24}$$

where $p(c_i)$ and $p(c'_i)$ are the probabilities that a document of the corpus belongs to cluster c_i and c'_i, respectively, and $p(c_i, c'_i)$ is the probability that the selected document belongs to c_i as well as c'_i at the same time. The MI information is then normalized with the following equation,

$$NMI(C, C') = \frac{MI(C, C')}{max(H(C), H(C'))} \tag{3.25}$$

where $H(C)$ and $H(C')$ are the entropies of C and C', respectively, This measure ranges from 0 to 1. When NMI is 1 the two clustering are identical, when it is 0, the two sets are independent.

3.6.1 Experiments with Dominant Set Clustering

The experiments with dominant set clustering are aimed at testing this framework in the document clustering task, a field in which, to the best of our knowledge, has never been tested. Furthermore, we want to test how this framework can be employed to identify only the most representative elements of a cluster and to propagate this information over the network, exploiting evolutionary dynamics, as described in Section 3.5.5. Furthermore, we tested our approaches with different similarity graphs, in order to identify which data representation better fits the algorithms.

3.6.1.1 Datasets

For the evaluation of our approach, we used the same datasets used in [175], where has been conducted an extensive comparison of different document clustering algorithms[1]. The test set is composed of 13 datasets, whose characteristics are illustrated in Table 3.1. It is composed of datasets of different sizes (n_d), from 204 documents (tr23) to 8580 (sports). The number of classes (K) of the datasets is different and ranges from 3 to 10. Another important characteristic of the datasets is the number of words (n_w) in the vocabulary of each dataset, which ranges from 5832 (tr23) to 41681 (classic) and is conditioned by the number of documents on the dataset and on the number of different topics in it. The last two features which describe the datasets are n_c and $Balance$. n_c represents the average number of documents per class and $Balance$ is the ratio among the number of documents in the smallest class and in the largest class.

3.6.1.2 Experiments with the Entire Feature Space

In this section we test our approach with the entire feature space of each dataset. The graphs for our experiments are prepared as described in Section 3.5.

[1]The datasets have been downloaded from, http://www.shi-zhong.com/software/docdata.zip . We noticed that the number of features on these datasets is lower that that indicated in [175], maybe due to some transcription errors. But we think that the results are not compromised, since the differences are really low, in the order of 1 to 100.

Data	Source	n_d	n_v	K	n_c	Balance
NG17-19	3 overlapping groups from NG20	2998	15810	3	999	0.998
classic	CACM/CISI/Cranfield/Medline	7094	41681	4	1774	0.323
k1b	WebACE	2340	21819	6	390	0.043
hitech	San Jose Mercury (TREC)	2301	10800	6	384	0.192
reviews	San Jose Mercury (TREC)	4069	18483	5	814	0.098
sports	San Jose Mercury (TREC)	8580	14870	7	1226	0.036
la1	LA Times (TREC)	3204	31472	6	534	0.290
la12	LA Times (TREC)	6279	31472	6	1047	0.282
la2	LA Times (TREC)	3075	31472	6	513	0.274
tr11	TREC	414	6424	9	46	0.046
tr23	TREC	204	5831	6	34	0.066
tr41	TREC	878	7453	10	88	0.037
tr45	TREC	690	8261	10	69	0.088

Table 3.1: Datasets description

	NG17-19	classic	k1b	hitech	review	sports	la1
AC	$.56 \pm 0$	$.66 \pm .07$	$.82 \pm 0$	$.44 \pm 0$	$.81 \pm 0$	$.69 \pm 0$	$.49 \pm .04$
MI	$.42 \pm 0$	$.56 \pm .22$	$.66 \pm 0$	$.27 \pm 0$	$.59 \pm 0$	$.62 \pm 0$	$.45 \pm .04$

Table 3.2: Results as accuracy (AC) and normalized mutual information (NMI), for the experiments on dominant set clustering with the entire feature space. Each experiment was run 50 times and is presented with standard deviation ($\pm$).

The results of these experiments are shown in Table 3.2 and Table 3.3 and will be used as point of comparison for the next experiments. The results do not show a stable pattern, in fact they range from MI .27 on the *hitech* dataset, to MI .67 on *k1b*. The reason of this incongruence is the representation of the datasets, which in some cases has no good discriminators for the described objects.

An example of the graphical representation of the two datasets mentioned above is presented in Figure 3.3, where we can see that the similarity matrices and the corresponding graphs constructed for *hitech* do not show a clear structure on the main diagonal. To the contrary, it is possible to recognize the cluster structures clearly in the graphs representing *k1b*.

3.6.1.3 Experiments with Basic Feature Selection

Each dataset described in [175], represents a corpus as BoW feature vectors, where each vector represents a document and each column indicates the number of occurrences of a particular word in the corresponding text. As we have seen in Section 2.5.4, this representation leads to high dimensional space. It gives to each feature

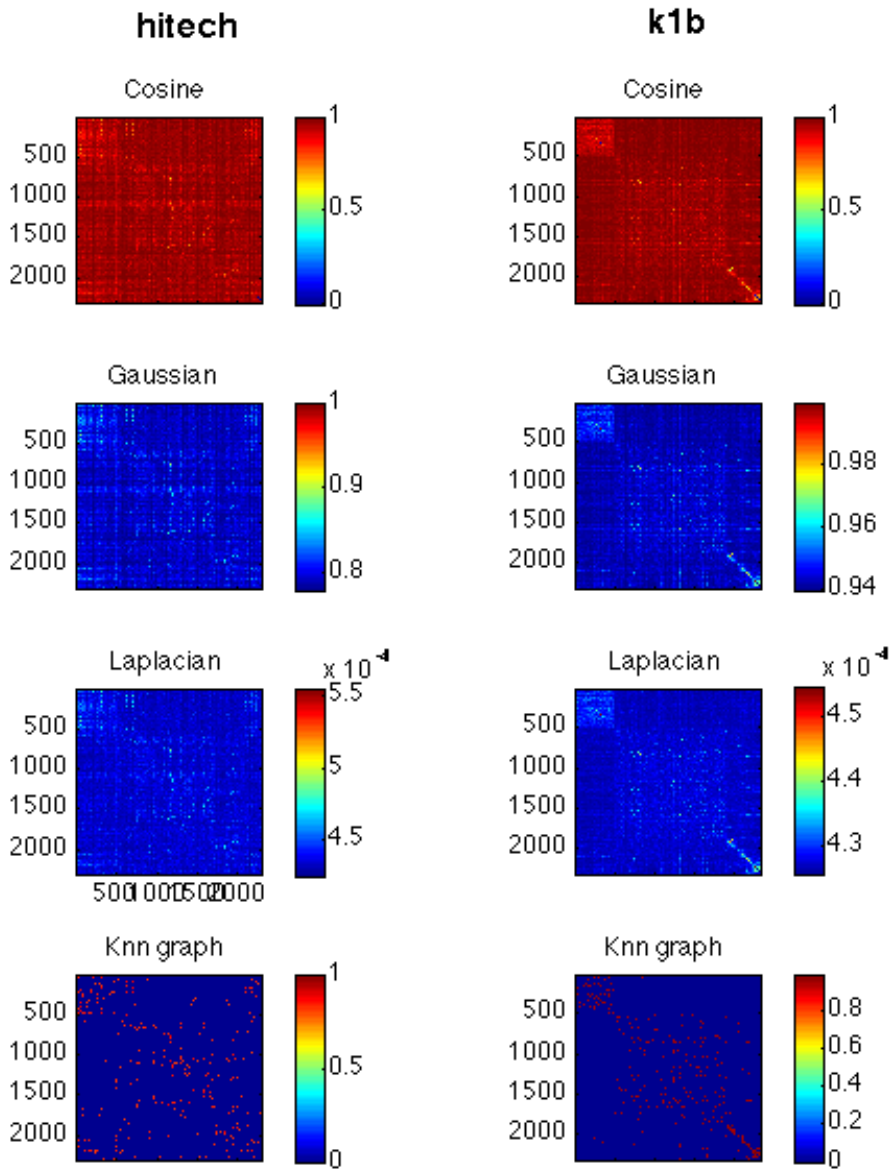


Figure 3.3: Different representations for the datasets *hitech* and *k1b*.

	la12	la2	tr11	tr23	tr41	tr45
AC	$.57 \pm .02$	$.54 \pm 0$	$.68 \pm .02$	$.44 \pm .01$	$.64 \pm .07$	$.64 \pm .02$
MI	$.46 \pm .01$	$.46 \pm .01$	$.63 \pm .02$	$.38 \pm 0$	$.53 \pm .06$	$.59 \pm .01$

Table 3.3: Results as accuracy (AC) and normalized mutual information (NMI), for the experiments on dominant set clustering with the entire feature space.Each experiment was run 50 times and is presented with standard deviation($\pm$).

	classic	k1b	la1	la12	la2
pre	41681	21819	31472	31472	31472
post	7616	10411	13195	17741	12432
%	0.82	0.52	0.58	0.44	0.6

Table 3.4: Number of features for each dataset before and after feature selection.

the same importance and does not take into account the problems of homonymy and synonymy. To overcome these limitations, we decided to apply to the corpora a basic frequency selection heuristic, which eliminates the features which occur more often than a determined thresholds. In this study only the words occurring more than once were kept.

This basic reduction leads to a more compact feature space, which is easier to handle. Words that appear very few times in the corpus can be special characters or miss-spelled words and for this reason can be eliminated. The number of features of the new dataset, after the frequency selection, are shown in Table 3.4. From the table, we can see that the reduction is significant for five of the datasets used, arriving at 82% of reduction for *classic*, the other datasets have not been affected by this process.

In Table 3.5 we show the results obtained employing the same algorithm used to test the datasets with all the features. This reduction can be considered a good choice to reduce the size of the datasets and the computational, but do not have a big impact on the performances of the algorithm. In fact, the results show that the improvements, in the performance of the algorithm, are not substantial. We have an improvement of 1%, in terms of NMI, in four datasets over five. In one dataset we obtained lower results. This could be due to the fact that we do not know exactly what words have been removed from the datasets, because they are not provided with the datasets. In fact, it is possible that the reduction has removed some important (discriminative) word from the feature space, compromising the representation of the documents.

	classic	k1b	la1	la12	la2
AC	$.67 \pm 0$	$.79 \pm 0$	$.56 \pm .11$	$.56 \pm .03$	$.57 \pm 0$
MI	$.57 \pm 0$	$.67 \pm 0$	$.47 \pm .12$	$.44 \pm .01$	$.47 \pm 0$

Table 3.5: Mean results as accuracy (AC) and normalized mutual information (NMI), for the experiments on dominant set clustering with frequency selection. Each experiment was run 50 times and is presented with standard deviation ($\pm$).

3.6.1.4 Experiments with Latent Semantic Analysis

In this section is presented the evaluation of the proposed approach, using LSA to construct a semantic space which reduces the dimensions of the feature space. The evaluation was conducted using different numbers of features to describe each dataset, ranging from 10 to 400. This is due to the fact that there is no agreement on the correct number of features to extract for a determined dataset. For this reason this value has to be calculate experimentally.

The results of this evaluation are shown in two different tables, Table 3.6 indicates the results as NMI and Table 3.7 indicates the results as accuracy. The performances of the algorithm measured as NMI are similar on average (excluding the case of 10 features), but there is no agreement on different datasets. In fact, different data representations affect heavily the performances on datasets such as NG17-19, where the performances ranges from .27 to .46. This phenomenon is due to the fact that each dataset has different characteristics, as shown in Table 3.1.

The results with this new representation of the data shows that the use of LSA is beneficial. In fact, it is possible to achieve results higher than with the entire feature space or with the frequency reduction. The improvements are substantial and in many cases are 10% higher.

3.6.1.5 Comparison to State-of-the-Art Algorithms

The results of the evaluation of the Document Clustering Games are shown in Table 3.8 and 3.9 (third column, DCG), where, for each dataset are compared the best results obtained with the document clustering games approach and the best results indicated in [175] and in [176]. In the first article was conducted an extensive evaluation of different generative and discriminative models, specifically tailored for document clustering and two graph-based approaches, CLUTO and a bipartite spectral co-clustering method, which obtained better performances than the other algorithms. The results in this article are reported as NMI. In the second article there is an evaluation on different NMF approaches to document clustering, on the same datasets that we used and the results are reported as AC.

From Table 3.8 it is possible to see that the results of the document clustering games are higher than those of state-of-the-art algorithms on ten datasets out of

Data	10	50	100	150	200	250	300	350	400
NG17-19	.27	.37	**.46**	.26	.35	.37	.36	.37	.37
classic	.53	.63	.71	.73	**.76**	.74	.72	.72	.69
k1b	**.68**	.61	.58	.62	.63	.63	.62	.61	.62
hitech	**.29**	.28	.25	.26	.28	.27	.27	.26	.26
reviews	**.60**	.59	.59	.59	.59	.59	.58	.58	.58
sports	.62	.63	**.69**	.67	.66	.66	.66	.64	.62
la1	.49	.53	.58	.58	.58	.57	**.59**	.57	**.59**
la12	.48	.52	.52	.52	.53	**.56**	.54	.55	.54
la2	.53	.56	.58	.58	.58	.58	**.59**	.58	.58
tr11	.69	.65	.67	.68	**.71**	.70	.70	.69	.70
tr23	.42	**.48**	.41	.39	.41	.40	.41	.40	.41
tr41	.65	.75	.72	.69	.71	.74	**.76**	.69	.75
tr45	.65	**.70**	.67	.69	.69	.68	.68	.67	.69
avg.	.53	.56	**.57**	.56	**.57**	**.57**	**.57**	.56	**.57**

Table 3.6: Results as normalized mutual information (NMI) for all the datasets. Each column indicates the results obtained with a reduced version of the feature space using LSA. Each experiment was run 50 times.

Data	10	50	100	150	200	250	300	350	400
NG17-19	.61	**.63**	.56	.57	.51	.51	.51	.51	.51
classic	.64	.76	.87	.88	**.91**	.88	.85	.84	.80
k1b	.72	.55	.58	.73	**.75**	**.75**	.73	.70	.73
hitech	**.48**	.36	.42	.41	.47	.46	.41	.43	.42
reviews	**.73**	.72	.69	.69	.69	.71	.71	.71	.71
sports	.62	.61	**.71**	.69	.68	.68	.68	.68	.61
la1	.59	.64	.72	.70	**.73**	.72	**.73**	.72	**.73**
la12	.63	.63	.62	.62	.63	**.67**	.64	**.67**	.65
la2	**.69**	.66	.60	.60	.61	.60	.65	.60	.60
tr11	.69	.66	.69	.70	**.72**	.71	.71	.71	.71
tr23	.44	**.51**	.43	.42	.43	.43	.43	.43	.43
tr41	.60	**.76**	.68	.68	.65	.75	.77	.67	.77
tr45	.57	**.69**	.66	.68	.67	.67	.67	.67	.67
avg.	.62	.63	.63	.64	.65	**.66**	.65	.64	.64

Table 3.7: Results as accuracy (AC) for all the datasets. Each column indicates the results obtained with a reduced version of the feature space using LSA. Each experiment was run 50 times.

Data	DCG_{noK}	DCG	$Best$
NG17-19	.39 ± 0	**.46** ± 0	**.46** ± .01
classic	.71 ± 0	**.76** ± 0	.71 ± .06
k1b	**.73** ± .02	.68 ± .02	.67 ± .04
hitech	**.35** ± .01	.29 ± .02	.33 ± .01
reviews	.57 ± .01	**.60** ± .01	.56 ± .09
sports	.67 ± 0	**.69** ± 0	.67 ± .01
la1	.53 ± 0	**.59** ± 0	.58 ± .02
la12	.52 ± 0	**.56** ± 0	**.56** ± .01
la2	.53 ± 0	**.59** ± 0	.56 ± .01
tr11	**.72** ± 0	.71 ± 0	.68 ± .02
tr23	**.57** ± .02	.48 ± .03	.43 ± .02
tr41	.70 ± .01	**.76** ± .06	.69 ± .02
tr45	**.70** ± .02	**.70** ± .03	.68 ± .05

Table 3.8: Results as NMI of generative models and graph partitioning algorithm ($Best$) compared to our approach with and without the number of clusters to extract. Each experiment was run 50 times.

thirteen. On the remaining three datasets we obtained the same results on two datasets and a lower result in one. On classic, tr23 and tr26 the improvement of our approach is substantial, with results higher than 5%. Form Table 3.9 we can see that our approach performs substantially better that NMF on all the datasets.

3.6.1.6 Experiments with no Class Number

The last experiment was conducted without using the number of clusters to extract. It has been tested the ability of dominant set to find natural clusters and the performances that can be obtained in this context by the document clustering games. In this way, we first run dominant set to discover many small clusters, setting the parameter of the gaussian kernel with a small value (0.1). Then we re-clusters the obtained clusters using as similarity matrix the similarities shared between the nodes of two different clusters.

The results of this evaluation are shown in Table 3.8 and 3.9 (second column, DCG_{noK}). The results show that this new formulation of the clustering games performs well in many datasets. In fact, in datasets such as k1b, hitech, tr11 and tr23 has results higher than the clustering games performed in the previous sections. This can be explained by the fact that with this formulation the number of clustered points is higher that in the previous version. This can improve the performances of the system when dominant set is able to find the exact number of natural clusters from the graph. To the contrary, when it not able to predict this number, the performances as NMI decrease drastically. This phenomenon can explain why in

Data	DCG_{noK}	DCG	$Best$
NG17-19	$.59 \pm 0$	$\mathbf{.63} \pm 0$	-
classic	$.80 \pm 0$	$\mathbf{.91} \pm 0$	$.59 \pm .07$
k1b	$\mathbf{.86} \pm .02$	$.75 \pm .03$	$.79 \pm 0$
hitech	$\mathbf{.52} \pm .01$	$.48 \pm .02$	$.48 \pm .04$
reviews	$.64 \pm .01$	$\mathbf{.73} \pm .01$	$.69 \pm .07$
sports	$\mathbf{.78} \pm 0$	$.71 \pm 0$	$.50 \pm .07$
la1	$.63 \pm 0$	$\mathbf{.73} \pm 0$	$.66 \pm 0$
la12	$.59 \pm 0$	$\mathbf{.67} \pm 0$	-
la2	$.55 \pm 0$	$\mathbf{.69} \pm 0$	$.53 \pm 0$
tr11	$\mathbf{.74} \pm 0$	$.72 \pm 0$	$.53 \pm .05$
tr23	$\mathbf{.52} \pm .02$	$.51 \pm .05$	$.43 \pm .06$
tr41	$.75 \pm .01$	$\mathbf{.76} \pm .08$	$.53 \pm .06$
tr45	$\mathbf{.71} \pm .01$	$.69 \pm .04$	$.54 \pm .06$

Table 3.9: Results as AC of NMF models ($Best$) compared to our approach with and without the number of clusters to extract. Each experiment was run 50 times.

some datasets it does not perform well. In fact, in datasets such as, NG18-19, la1, la12 and l2 the performances of the system are very low.

3.6.2 Experiments with NMF

As introduced in Section 3.5, we used the matrix U, obtained by applying the NMF technique on the *tf-idf* matrix of each corpus, to define the inclination that each document has toward a determined cluster. Therefore, we initialized the strategy space of each player according to this information. After this system initialization, we started the dynamics of the document clustering games, using different k-nn graphs, constructed from the proximity matrix obtained with the classic cosine distance.

3.6.2.1 Datasets

We conducted an extensive evaluation of our approach using the Reuters-21578[2] corpus, a common benchmark for the evaluation of clustering and classification algorithms. It is composed of 21578 documents divided in 135 classes. Each document is a news which appeared in the Reuters newswire in 1987. The news have been manually indexed and associated to one or more classes by personnel from Reuters Ltd. and Carnegie Group, Inc. and made available in 1990.

For our experiments we excluded documents which have more than one class and classes which contain less than 6 documents. The final dataset which we have used is composed of 9455 documents divided in 50 classes. The distribution of documents

[2]The corpus has been downloaded from http://kdd.ics.uci.edu/databases/reuters21578

	Accuracy				NMI			
K	NMF	*cosine*	*gaus*	*lapl*	NMF	*cosine*	*gaus*	*lapl*
2	.94	.94	.94	.94	.72	.73	.74	.74
3	.88	.89	.89	.89	.73	.75	.76	.76
4	.83	.84	.84	.84	.69	.71	.71	.71
5	.75	.77	.77	.77	.64	.65	.66	.65
6	.72	.76	.76	.76	.64	.65	.67	.66
7	.72	.75	.75	.75	.65	.66	.67	.66
8	.70	.75	.75	.75	.66	.67	.67	.66
9	.65	.72	.72	.72	.61	.60	.61	.61
10	.61	.66	.67	.67	.58	.54	.55	.55
Avg	.76	**.79**	**.79**	**.79**	.66	.66	**.67**	**.67**

Table 3.10: Results as accuracy and NMI comparing the results obtained with NMF and document clustering games with different similarity graphs: simple cosine similarity, gaussian kernel and Laplacian. For each number of classes K the experiments were run on 50 different datasets.

per class is very skewed, it ranges from 6 to 3944 and constitutes a big challenge for any clustering algorithm, when it is evaluated using the NMI, since it takes into account the purity of the clusters extracted and not just the number of points clustered correctly.

Once we have redefined the entire dataset, we have created, from it, new datasets, composed of different number of classes, ranging from 2 to 10. For each different number of classes we created 50 randomly selected datasets, in order to evaluate our approach in different situations.

3.6.2.2 Performance Evaluations and Comparisons

The results of the evaluation with NMF are shown in table 3.10. This experiment was conducted using three different similarity graphs to model the interaction between the players: the cosine similarity, the gaussian kernel and the gaussian kernel with normalized laplacian. We used these similarity measure to weight the edges of the graph and then reduced the number of edges keeping only the five most similar nodes for each vertex in the graph.

As it is possible to see from the table, the results of this approach is particularly effective in terms of accuracy. In fact, it is possible to obtain an improvement of 3% on average. The improvement as NMI is not high. This is due to the fact that the datasets are very unbalanced. In fact it is possible to have a cluster of 5 points and one of thousands of points in the same dataset and the inaccuracy in clustering the small dataset heavily affect the performances of the system.

3.7 Conclusions

In this chapter we have explored new methods for document clustering based on game theory. We have provided an introduction on the document clustering task, highlighting its challenges and explaining the motivations behind the choice a game theoretic framework. We have conducted an extensive series of experiments to test the approach on different scenarios. We have also evaluated the system with different implementations and compared the results with state-of-the-art algorithms.

Our method can be considered as a continuation of graph based approaches but it combines together the partition of the graph and the propagation of the information across the network. With this method we used the structural information about the graph and then we employed evolutionary dynamics to find the best labeling of the data points. The application of a game theoretic framework is able to exploit relational and contextual information and guarantees that the final labeling is consistent.

The system has demonstrated to perform well compared with state-of-the-art system and to be extremely flexible. In fact, it is possible to implement new graph similarity measure or new dynamics to improve the results or to adapt it to different contexts.

The work with dominant set demonstrated that it is possible to use few labeled nodes to label an entire graph. The work on non-negative matrix factorization demonstrated that the results of that is possible to use the information derived from this technique to obtain better clusterings. This because our approach exploit relational information among data points and try to find stable correlations.

Conclusions

This dissertation has contributed to the vast field of game theoretic models of learning. An attempt has been made to give a biological explanation, to the task of computational learning, using concepts derived from evolutionary game theory. Learning was treated a an ability, which naturally emerges from the observation of the contextual environment. In this perspective, we interpreted the learning process as a dynamical system, in which the parts of the system have to learn from each other, interacting and exchanging information. These interactions allow the player to gradually change, leading to a progressive transformation of the entire system. Each time that the system evolves, the traits of each part became clearer, moving from a state in which there is ambiguity to a state in which each part knows what is its nature.

This work was focused problems related to classification and clustering, adopting graph theoretic principles to model the geometry of the data and game theoretic frameworks to find the best class assignments, exploiting relational and contextual information. This model has shown to be useful and versatile, because it can be employed as an unsupervised, semi-supervised or supervised learning model. Furthermore, it is possible to use it adopting many state.of-the-art tools for the computation of pairwise-similarity among objects.

The experiments on word sense disambiguation have shown that our approach is able to perform a consistent labeling of the data, We demonstrated that our formulation of the problem is beneficial for this task, because state-of-the-art systems do not have a notion of consistency when they try to disambiguate the words, instead they just try to find the most important sense in a network.

The work on document clustering has tested how dominant set clustering works on textual data and have shown different perspectives on how the partial results of this algorithm can be used to cluster an entire graph. This turns out to be particularly important in case of large graphs, which would be difficult to treat or in the case of dynamical graphs, which grow over time. In fact, with this approach it is possible to use the data points at time t to cluster incoming data at time $t+1$.

The contribution of the work on NMF was to explore new methodologies to refine the clustering obtained with this technique. We demonstrated how it is possible to obtain better results with our formulation, overcoming the simplistic approach of maximization proposed in literature. This improvement can be considered as substantial, since An important contribution since NMF is largely used in different fields of data analysis.

The overall contribution of this thesis is that it shows how game-theoretic models can be employed for different natural language processing tasks. We have opened a

new perspective in this discipline and explored different techniques in game theory for classification and clustering, demonstrating that these techniques can have a large number of applications arising from a variety of fields. As future work we are planning to extend the use of the methodologies described in this thesis to other field and to explore new game settings and dynamics.

Bibliography

[1] Jasper Snoek, Hugo Larochelle, and Ryan P Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in neural information processing systems*, pages 2951–2959, 2012.

[2] J. Nash. Non-cooperative games. *Annals of mathematics*, pages 286-295, 1951.

[3] J Maynard Smith and GR Price. The logic of animal conflict. *Nature*, 246:15, 1973.

[4] Miller McPherson, Lynn Smith-Lovin, and James M Cook. Birds of a feather: Homophily in social networks. *Annual review of sociology*, pages 415–444, 2001.

[5] Massimiliano Pavan and Marcello Pelillo. Dominant sets and pairwise clustering. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(1):167–172, 2007.

[6] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[7] Charles Hartshorne, Paul Weiss, and Arthur Burks. *Collected Papers of Charles Sanders Peirce*, volume 1. Harvard University Press, 1967.

[8] F De Saussure. 1966. Course in general linguistics, 1916.

[9] Charles Darwin. *The descent of man, and selection in relation to sex. By Charles Darwin ...*, volume 2. London,J. Murray,, 1871. http://www.biodiversitylibrary.org/bibliography/2092.

[10] Ludwig Wittgenstein. Philosophical investigations. 1967. *Oxford: Blackwell*, 8:250, 1953.

[11] Yuri Lin, Jean-Baptiste Michel, Erez Lieberman Aiden, Jon Orwant, Will Brockman, and Slav Petrov. Syntactic annotations for the google books ngram corpus. In *Proceedings of the ACL 2012 system demonstrations*, pages 169–174. Association for Computational Linguistics, 2012.

[12] Morten H Christiansen and Simon Kirby. Language evolution: Consensus and controversies. *Trends in cognitive sciences*, 7(7):300–307, 2003.

[13] Claudio Castellano, Santo Fortunato, and Vittorio Loreto. Statistical physics of social dynamics. *Reviews of modern physics*, 81(2):591, 2009.

[14] Henri Poincaré. Sur le probleme des trois corps et les équations de la dynamique. *Acta mathematica*, 13(1):A3–A270, 1890.

[15] David Lewis. *Convention: A philosophical study*. Harvard University Press, 1967.

[16] Simon M Huttegger and Kevin JS Zollman. Signaling games. In *Language, games, and evolution*, pages 160–176. Springer, 2011.

[17] Brian Skyrms. *Evolution of the Social Contract*. Cambridge University Press, 1996.

[18] Brian Skyrms. Evolution of inference. *Dynamics of human and primate societies*, pages 77–88, 2000.

[19] Martin A Nowak. The basic reproductive ratio of a word, the maximum size of a lexicon. *Journal of theoretical biology*, 204(2):179–189, 2000.

[20] Martin A Nowak, Natalia L Komarova, and Partha Niyogi. Computational and evolutionary aspects of language. *Nature*, 417(6889):611–617, 2002.

[21] Natalia L Komarova, Partha Niyogi, and Martin A Nowak. The evolutionary dynamics of grammar acquisition. *Journal of theoretical biology*, 209(1):43–59, 2001.

[22] Luc Steels. The origins of syntax in visually grounded robotic agents. *Artificial Intelligence*, 103(1):133–156, 1998.

[23] William S-Y Wang, Jinyun Ke, and James W Minett. Computational studies of language evolution. *Lang. Ling. Monograph Series B*, pages 65–108, 2004.

[24] Thorsten Brants and Alex Franz. {Web 1T 5-gram Version 1}. 2006.

[25] Geoffrey Leech. 100 million words of english: the british national corpus (bnc). *Language Research*, 28(1):1–13, 1992.

[26] Luigi Luca Cavalli-Sforza and Marcus W Feldman. *Cultural transmission and evolution: a quantitative approach*. Number 16. Princeton University Press, 1981.

[27] Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K Gray, Joseph P Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, et al. Quantitative analysis of culture using millions of digitized books. *science*, 331(6014):176–182, 2011.

[28] Drew Fudenberg and David K Levine. *The theory of learning in games*, volume 2. MIT press, 1998.

[29] Carsten Ullrich. Descriptive and prescriptive learning theories. In *Pedagogically Founded Courseware Generation for Web-Based Learning*, pages 37–42. Springer, 2008.

[30] Stathis Grigoropoulos. The minority game: Individual and social learning. 2014.

[31] Fernando Vega-Redondo. *Economics and the Theory of Games*. Cambridge university press, 2003.

[32] Martin A Nowak and Karl Sigmund. Evolutionary dynamics of biological games. *science*, 303(5659):793–799, 2004.

[33] Bryan D Jones. Bounded rationality. *Annual review of political science*, 2(1):297–321, 1999.

[34] J. Von Neumann and O. Morgenstern. *Theory of Games and Economic Behavior (60th Anniversary Commemorative Edition)*. Princeton University Press, 1944.

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

[36] Samir Okasha and Ken Binmore. *Evolution and rationality: decisions, cooperation and strategic behaviour*. Cambridge University Press, 2012.

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

[38] Jörgen W Weibull. *Evolutionary game theory*. MIT press, 1997.

[39] William H Sandholm. *Population games and evolutionary dynamics*. MIT press, 2010.

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

[41] W. Weaver. Translation. *Machine translation of languages*, 14:15-23, 1955.

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

[43] I. Dagan and O. Glickman. Probabilistic textual entailment: Generic applied modeling of language variability. 2004.

[44] D. Vickrey, L. Biewald, M. Teyssier, and D. Koller. 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. Association for Computational Linguistics, 2005.

[45] P. Smrž. Using wordnet for opinion mining. In *Proceedings of the Third International WordNet Conference*, pages 333-335. Masaryk University, 2006.

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

[47] Zhi Zhong and Hwee Tou Ng. 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. Association for Computational Linguistics, 2012.

[48] P. Pantel and D. Lin. Discovering word senses from text. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 613-619. ACM, 2002.

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

[50] Stephen Tratz, Antonio Sanfilippo, Michelle Gregory, Alan Chappell, Christian Posse, and Paul Whitney. Pnnl: A supervised maximum entropy approach to word sense disambiguation. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 264–267. Association for Computational Linguistics, 2007.

[51] Thanh Phong Pham, Hwee Tou Ng, and Wee Sun Lee. Word sense disambiguation with semi-supervised learning. In *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, volume 20, page 1093. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2005.

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

[53] Rada Mihalcea. 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. Association for Computational Linguistics, 2005.

[54] Diana McCarthy, Rob Koeling, Julie Weeds, and John Carroll. Unsupervised acquisition of predominant word senses. *Computational Linguistics*, 33(4):553–590, 2007.

[55] Simone Paolo Ponzetto and Roberto Navigli. Knowledge-rich word sense disambiguation rivaling supervised systems. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 1522–1531. Association for Computational Linguistics, 2010.

[56] Eneko Agirre, Oier Lopez De Lacalle, Aitor Soroa, and Informatika Fakultatea. Knowledge-based wsd and specific domains: Performing better than generic supervised wsd. In *IJCAI*, pages 1501–1506, 2009.

[57] Ravi Som Sinha and Rada Mihalcea. Unsupervised graph-basedword sense disambiguation using measures of word semantic similarity. In *ICSC*, volume 7, pages 363–369, 2007.

[58] Rada Mihalcea. Knowledge-based methods for wsd. *Word Sense Disambiguation: Algorithms and Applications*, pages 107–131, 2006.

[59] Roberto Navigli and Mirella Lapata. Graph connectivity measures for unsupervised word sense disambiguation. In *IJCAI*, pages 1683–1688, 2007.

[60] Eneko Agirre, David Martínez, Oier López de Lacalle, and Aitor Soroa. 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. Association for Computational Linguistics, 2006.

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

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

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

[64] Jin Cong and Haitao Liu. Approaching human language with complex networks. *Physics of life reviews*, 11(4):598–618, 2014.

[65] Diane Larsen-Freeman and Lynne Cameron. *Complex systems and applied linguistics*. Oxford University Press, 2008.

[66] Devendra Singh Chaplot, Pushpak Bhattacharyya, and Ashwin Paranjape. Unsupervised word sense disambiguation using markov random field and dependency parser. 2015.

[67] Jon Kleinberg and Eva Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. *Journal of the ACM (JACM)*, 49(5):616–639, 2002.

[68] J. R. Firth. A synopsis of linguistic theory 1930-1955. *Studies in linguistic analysis. Oxford: Blackwell*, 1957.

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

[70] David Yarowsky. One sense per collocation. In *Proceedings of the workshop on Human Language Technology*, pages 266–271. Association for Computational Linguistics, 1993.

[71] Ahti-Veikko Pietarinen. *Game theory and linguistic meaning*. BRILL, 2007.

[72] Brian Skyrms. *Signals: Evolution, learning, and information*. Oxford University Press, 2010.

[73] Martin A Nowak, Natalia L Komarova, and Partha Niyogi. Evolution of universal grammar. *Science*, 291(5501):114–118, 2001.

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

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

[76] James R Curran, Tara Murphy, and Bernhard Scholz. Minimising semantic drift with mutual exclusion bootstrapping. In *Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics*, volume 6, 2007.

[77] Rocco Tripodi, Marcello Pelillo, and Rodolfo Delmonte. An evolutionary game theoretic approach to word sense disambiguation. In *Natural Language Processing and Cognitive Science*, pages 39–48. DE GRUYTER, 2015.

[78] Rocco Tripodi and Marcello Pelillo. Wsd-games: a game-theoretic algorithm for unsupervised word sense disambiguation. In *Proceedings of SemEval-2015.*, 2015.

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

[80] George A Miller, Claudia Leacock, Randee Tengi, and Ross T Bunker. A semantic concordance. In *Proceedings of the workshop on Human Language Technology*, pages 303–308. Association for Computational Linguistics, 1993.

[81] Hwee Tou Ng and Hian Beng Lee. Integrating multiple knowledge sources to disambiguate word sense: An exemplar-based approach. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*, pages 40–47. Association for Computational Linguistics, 1996.

[82] Yee Seng Chan and Hwee Tou Ng. Scaling up word sense disambiguation via parallel texts. In *AAAI*, volume 5, pages 1037–1042, 2005.

[83] David Yarowsky and Radu Florian. Evaluating sense disambiguation across diverse parameter spaces. *Natural Language Engineering*, 8(04):293–310, 2002.

[84] Rob Koeling, Diana McCarthy, and John Carroll. 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. Association for Computational Linguistics, 2005.

[85] Michael Lesk. 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. ACM, 1986.

[86] Philip Resnik. Using information content to evaluate semantic similarity in a taxonomy. *arXiv preprint cmp-lg/9511007*, 1995.

[87] Siddharth Patwardhan, Satanjeev Banerjee, and Ted Pedersen. Using measures of semantic relatedness for word sense disambiguation. In *Computational linguistics and intelligent text processing*, pages 241–257. Springer, 2003.

[88] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. 1999.

[89] Rada Mihalcea, Paul Tarau, and Elizabeth Figa. Pagerank on semantic networks, with application to word sense disambiguation. In *Proceedings of the 20th international conference on Computational Linguistics*, page 1126. Association for Computational Linguistics, 2004.

[90] Taher H Haveliwala. Topic-sensitive pagerank. In *Proceedings of the 11th international conference on World Wide Web*, pages 517–526. ACM, 2002.

[91] Diego De Cao, Roberto Basili, Matteo Luciani, Francesco Mesiano, and Riccardo Rossi. 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. Association for Computational Linguistics, 2010.

[92] Steve L Manion and Raazesh Sainudiin. An iterative 'sudoku style'approach to subgraph-based word sense disambiguation. *Lexical and Computational Semantics (* SEM 2014)*, page 40, 2014.

[93] Michael I Jordan and Yair Weiss. Probabilistic inference in graphical models. *Handbook of neural networks and brain theory*, 2002.

[94] Roberto Navigli and Simone Paolo Ponzetto. 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. Association for Computational Linguistics, 2012.

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

[96] Roberto Navigli and Simone Paolo Ponzetto. Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. *Artificial Intelligence*, 193:217–250, 2012.

[97] Christian Bizer, Tom Heath, and Tim Berners-Lee. Linked data-the story so far. *Semantic Services, Interoperability and Web Applications: Emerging Concepts*, pages 205–227, 2009.

[98] Hanghang Tong, Christos Faloutsos, and Jia-Yu Pan. Fast random walk with restart and its applications. 2006.

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

[100] Mohamed El Bachir Menai. Word sense disambiguation using evolutionary algorithms–application to arabic language. *Computers in Human Behavior*, 41:92–103, 2014.

[101] John H Holland. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.* U Michigan Press, 1975.

[102] Pablo Moscato et al. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. *Caltech concurrent computation program, C3P Report*, 826:1989, 1989.

[103] Didier Schwab, Andon Tchechmedjiev, Jérôme Goulian, Mohammad Nasiruddin, Gilles Sérasset, and Hervé Blanchon. Getalp: Propagation of a lesk measure through an ant colony algorithm. *Atlanta, Georgia, USA*, page 232, 2013.

[104] Edward Tsang. Foundations of constraint satisfaction. 1995.

[105] Douglas A Miller and Steven W Zucker. Copositive-plus lemke algorithm solves polymatrix games. *Operations Research Letters*, 10(5):285–290, 1991.

[106] Claudio Carpineto and Giovanni Romano. A survey of automatic query expansion in information retrieval. *ACM Computing Surveys (CSUR)*, 44(1):1, 2012.

[107] Fethi Fkih and Mohamed Nazih Omri. Learning the size of the sliding window for the collocations extraction: a roc-based approach. In *Proc. The 2012 International Conference on Artificial Intelligence (ICAI'12), Las Vegas, USA*, pages 1071–1077, 2012.

[108] Sandra Kübler, Ryan McDonald, and Joakim Nivre. Dependency parsing. *Synthesis Lectures on Human Language Technologies*, 1(1):1–127, 2009.

[109] Marie-Catherine De Marneffe, Bill MacCartney, Christopher D Manning, et al. Generating typed dependency parses from phrase structure parses. In *Proceedings of LREC*, volume 6, pages 449–454, 2006.

[110] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39-41, 1995.

[111] Mihoko Kitamura and Yuji Matsumoto. Automatic extraction of word sequence correspondences in parallel corpora. In *Proceedings of the 4th Workshop on Very Large Corpora*, pages 79–87, 1996.

[112] Siddharth Patwardhan and Ted Pedersen. 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. Citeseer, 2006.

[113] Martha Palmer, Christiane Fellbaum, Scott Cotton, Lauren Delfs, and Hoa Trang Dang. English tasks: All-words and verb lexical sample. In *The Proceedings of the Second International Workshop on Evaluating Word Sense Disambiguation Systems*, pages 21–24. Association for Computational Linguistics, 2001.

[114] Benjamin Snyder and Martha Palmer. 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, 2004.

[115] Sameer S Pradhan, Edward Loper, Dmitriy Dligach, and Martha Palmer. Semeval-2007 task 17: English lexical sample, srl and all words. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 87–92. Association for Computational Linguistics, 2007.

[116] Roberto Navigli, Kenneth C Litkowski, and Orin Hargraves. Semeval-2007 task 07: Coarse-grained english all-words task. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 30–35. Association for Computational Linguistics, 2007.

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

[118] Johannes Hoffart, Stephan Seufert, Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. Kore: keyphrase overlap relatedness for entity disambiguation. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 545–554. ACM, 2012.

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

[120] Kenneth Ward Church and Patrick Hanks. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29, 1990.

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

[122] Don Blaheta and Mark Johnson. Unsupervised learning of multi-word verbs. In *Proc. of the ACL/EACL 2001 Workshop on the Computational Extraction, Analysis and Exploitation of Collocations*, pages 54–60, 2001.

[123] CR Rao. Karl pearson chi-square test the dawn of statistical inference. In *Goodness-of-fit tests and model validity*, pages 9–24. Springer, 2002.

[124] Morris H DeGroot, Mark J Schervish, Xiangzhong Fang, Ligang Lu, and Dongfeng Li. *Probability and statistics*, volume 2. Addison-Wesley Reading, MA, 1986.

[125] Stefan Evert. Corpora and collocations. *Corpus Linguistics. An International Handbook*, 2:223–233, 2008.

[126] Rada Mihalcea and Dan I Moldovan. extended wordnet: Progress report. In *in Proceedings of NAACL Workshop on WordNet and Other Lexical Resources*. Citeseer, 2001.

[127] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of the 32nd annual meeting on Association for Computational Linguistics*, pages 133–138. Association for Computational Linguistics, 1994.

[128] Jay J Jiang and David W Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *arXiv preprint cmp-lg/9709008*, 1997.

[129] Ted Pedersen. Duluth: Measuring degrees of relational similarity with the gloss vector measure of semantic relatedness. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 497–501. Association for Computational Linguistics, 2012.

[130] José Camacho-Collados, Mohammad Taher Pilehvar, and Roberto Navigli. Nasari: a novel approach to a semantically-aware representation of items. In *Proceedings of NAACL*, pages 567–577, 2015.

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

[132] Steven Bird. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics, 2006.

[133] Eneko Agirre and Philip Glenny Edmonds. *Word sense disambiguation: Algorithms and applications*, volume 33. Springer Science & Business Media, 2007.

[134] Hoa Trang Dang. Investigations into the role of lexical semantics in word sense disambiguation. 2004.

[135] Ingo Steinwart and Andreas Christmann. *Support vector machines*. Springer Science & Business Media, 2008.

[136] Carlo Strapparava, Alfio Gliozzo, and Claudio Giuliano. Pattern abstraction and term similarity for word sense disambiguation: Irst at senseval-3. In *Proc. of SENSEVAL-3 Third International Workshop on Evaluation of Systems for the Semantic Analysis of Text*, pages 229–234. Citeseer, 2004.

[137] Johannes Hoffart, Fabian M Suchanek, Klaus Berberich, and Gerhard Weikum. 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. AAAI Press, 2013.

[138] Claude Sammut and Geoffrey I Webb. *Encyclopedia of machine learning*. Springer Science & Business Media, 2011.

[139] Michael R Anderberg. Cluster analysis for applications. *Probability and Mathematical Statistics, New York: Academic Press, 1973*, 1, 1973.

[140] Eliezer Geisler. *Knowledge and Knowledge Systems: Learning from the Wonders of the Mind: Learning from the Wonders of the Mind*. IGI Global, 2007.

[141] Eleanor Rosch and Barbara B Lloyd. Cognition and categorization. *Hillsdale, New Jersey*, 1978.

[142] Eleanor H Rosch. Natural categories. *Cognitive psychology*, 4(3):328–350, 1973.

[143] Anil K Jain, M Narasimha Murty, and Patrick J Flynn. Data clustering: a review. *ACM computing surveys (CSUR)*, 31(3):264–323, 1999.

[144] Harold Edson Driver and Alfred Louis Kroeber. *Quantitative expression of cultural relationships*. University of California Press, 1932.

[145] Robert Choate Tryon. *Cluster analysis: correlation profile and orthometric (factor) analysis for the isolation of unities in mind and personality*. Edwards brother, Incorporated, lithoprinters and publishers, 1939.

[146] Raymond Bernard Cattell. The description of personality: basic traits resolved into clusters. *The journal of abnormal and social psychology*, 38(4):476, 1943.

[147] Robert R Sokal. The principles and practice of numerical taxonomy. *Taxon*, pages 190–199, 1963.

[148] Rui Xu, Donald Wunsch, et al. Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678, 2005.

[149] Charles T Zahn. Graph-theoretical methods for detecting and describing gestalt clusters. *Computers, IEEE Transactions on*, 100(1):68–86, 1971.

[150] Zhenyu Wu and Richard Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(11):1101–1113, 1993.

[151] J Gary Augustson and Jack Minker. An analysis of some graph theoretical cluster techniques. *Journal of the ACM (JACM)*, 17(4):571–588, 1970.

[152] Vijay V Raghavan and CT Yu. A comparison of the stability characteristics of some graph theoretic clustering methods. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (4):393–402, 1981.

[153] Jagan Sankaranarayanan, Hanan Samet, Benjamin E Teitler, Michael D Lieberman, and Jon Sperling. Twitterstand: news in tweets. In *Proceedings of the 17th acm sigspatial international conference on advances in geographic information systems*, pages 42–51. ACM, 2009.

[154] Krishna Bharat, Michael Curtiss, and Michael Schmitt. Methods and apparatus for clustering news content, July 28 2009. US Patent 7,568,148.

[155] Mariona Coll Ardanuy and Caroline Sporleder. Structure-based clustering of novels. *EACL 2014*, pages 31–39, 2014.

[156] Inderjit S Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 269–274. ACM, 2001.

[157] Christopher D Manning, Prabhakar Raghavan, Hinrich Schütze, et al. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.

[158] Loulwah AlSumait and Carlotta Domeniconi. Text clustering with local semantic kernels. In *Survey of Text Mining II*, pages 87–105. Springer, 2008.

[159] George Kingsley Zipf. The psycho-biology of language. 1935.

[160] George Kingsley Zipf. Human behavior and the principle of least effort. 1949.

[161] Thomas K Landauer, Peter W Foltz, and Darrell Laham. An introduction to latent semantic analysis. *Discourse processes*, 25(2-3):259–284, 1998.

[162] Steven P Crain, Ke Zhou, Shuang-Hong Yang, and Hongyuan Zha. Dimensionality reduction and topic modeling: From latent semantic indexing to latent dirichlet allocation and beyond. In *Mining text data*, pages 129–161. Springer, 2012.

[163] Thomas Hofmann. Probabilistic latent semantic indexing. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pages 50–57. ACM, 1999.

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

[165] Wei Xu, Xin Liu, and Yihong Gong. Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 267–273. ACM, 2003.

[166] Chris Ding, Tao Li, and Wei Peng. Nonnegative matrix factorization and probabilistic latent semantic indexing: Equivalence chi-square statistic, and a hybrid method. In *Proceedings of the national conference on artificial intelligence*, volume 21, page 342. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2006.

[167] Ying Zhao and George Karypis. Empirical and theoretical comparisons of selected criterion functions for document clustering. *Machine Learning*, 55(3):311–331, 2004.

[168] Anil K Jain and Richard C Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[169] John Shawe-Taylor and Nello Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[170] Simon Haykin and Neural Network. A comprehensive foundation. *Neural Networks*, 2(2004), 2004.

[171] Anna Dagmar Peterson. A separability index for clustering and classification problems with applications to cluster merging and systematic evaluation of clustering algorithms. 2011.

[172] Josef Hofbauer and Karl Sigmund. Evolutionary game dynamics. *Bulletin of the American Mathematical Society*, 40(4):479–519, 2003.

[173] L Lovasz. Matching theory (north-holland mathematics studies). 1986.

[174] Alexander Strehl and Joydeep Ghosh. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *The Journal of Machine Learning Research*, 3:583–617, 2003.

[175] Shi Zhong and Joydeep Ghosh. Generative model-based document clustering: a comparative study. *Knowledge and Information Systems*, 8(3):374–384, 2005.

[176] Filippo Pompili, Nicolas Gillis, P-A Absil, and François Glineur. Two algorithms for orthogonal nonnegative matrix factorization with application to clustering. *Neurocomputing*, 141:15–25, 2014.