Morphological Annotation of a Corpus with a Collaborative Multiplayer Game

Onur Güngör and Tunga Güngör

Department of Computer Engineering, Boğaziçi University,
Bebek, 34342 Istanbul,Turkey
{onurgu,gungort}@boun.edu.tr

Abstract. In most of the natural language processing tasks, state-of-the-art systems usually rely on machine learning methods for building their mathematical models. Given that the majority of these systems employ supervised learning strategies, a corpus that is annotated for the problem area is essential. The current method for annotating a corpus is to hire several experts and make them annotate the corpus manually or by using a helper software. However, this method is costly and time-consuming. In this paper, we propose a novel method that aims to solve these problems. By employing a multiplayer collaborative game that is playable by ordinary people on the Internet, it seems possible to direct the covert labour force so that people can contribute by just playing a fun game. Through a game site which incorporates some functionality inherited from social networking sites, people are motivated to contribute to the annotation process by answering questions about the underlying morphological features of a target word. The experiments show that the 63.5% of the actual question types are successful based on a two-phase evaluation.

1 Introduction

In most of the natural language processing tasks, state-of-the-art systems usually rely on machine learning methods for building their mathematical models [1]. Given that the majority of these systems employ supervised learning strategies, a corpus that is annotated for the problem area is essential.

But having a relevantly annotated corpus is not enough on its own. The corpus must have a number of crucial features. First, it must include a set of carefully selected examples so that the method can train the model without bias. For the training to be successful, the corpus must include a set of examples. The size of the set is mainly determined by the characteristics of the training method itself. In addition to be sufficient for training, the corpus must not introduce bias to the trained model. Second, the corpus must be free of errors. While some methods may be resistant to several kinds of errors in the corpus, in most cases the errors prevent the method from training the model to its maximum extent.

When we recognize the crucial value of an error-free corpus with a vast number of examples in solving natural language processing tasks, the task of building a corpus with these properties gains importance. The most prominent method of building corpora today is to divide the work among experts and wait for them to finish their work [2]. However, it can be argued that this method is flawed in a number of points. First of all, this method dictates that the people who work on the work units must be experts in their field. Furthermore, they must be trained for this task. However, finding and training an expert is costly and time consuming. Even if we were successful in finding and hiring experts to work on building the corpus, there are other things that hinder the process. For example, the annotation patterns of two experts -even if they are highly experienced in the area- may be very different resulting in inconsistent annotation. We can expect to observe this situation especially in small and spontaneous annotation projects, where experts do not work in pairs and do not correct inconsistencies introduced by other experts later.

As a result of these problems, the process of building a corpus with the current methods is slow and expensive, if not low quality. This in turn impacts the rate of natural language processing research as well as its scope. This paper recognizes this problem as an important hindrance to the further development of natural language processing research and proposes a new method for building corpora.

We chose the morphological disambiguation of Turkish as the target domain. Morphological disambiguation problem is to select the correct morphological parse of a word in a given context among all of the possible parses of a word. We had two reasons for selecting this domain. First, this problem is at the core of other Turkish natural language processing tasks, i.e. parsing, speech recognition and sense tagging to name a few. Second, we have access to a corpus already tagged, which enabled us to test our results. In fact, the annotated corpus [3] is one of the very few annotated corpora in Turkish.

In this paper, we propose a novel system which incorporates a collaborative game for the morphological disambiguation of Turkish language. The game addresses the issues stated above and has two modes, one with a single player, where quiz-like questions are answered; the second is a two person game where one tries to explain a concealed word to the other, meanwhile answer some questions that are valuable for our annotation needs. The game is open to anyone and hosted on a publicly accessible web server.

We continue with the related literature on the subject in Section 2. Section 3 describes the game and the overall system that encapsulates the game in detail. Section 4 describes the experiment's setup and the obtained results. In Section 5, we draw conclusions and discuss some further research topics to be pursued.

2 Related Work

In [4], a game in which players are matched up with each other randomly and expected to win points by matching their inputs when viewing the same image

simultaneously is described. Given that no other means of communication is possible, the most obvious thing to input is the most distinctive figure in the image. It posed a nice challenge, this caused people to have a lot of fun and some of them eventually grew an addiction which lead to a very effective and fast way of labeling images on the web. This is the seminal work which introduced the idea of turning particular problems into games that people enjoy by harvesting the "wasted human-cycles"[1].

Later games by Luis von Ahn further extended the idea to various areas. Peekaboom [5] utilizes the idea to mark the portions of the images that depict target labels. Phetch [6] collects text descriptions of images by making one player describe the image and a group of players to simultaneously guess from the set of images they are confronted by a search engine result. Verbosity [7] collects facts about objects again by exploiting the collaborative game play method explained before. In Verbosity, one player tries to get the other player to guess the secret word that is exposed to her. Clues to the other user are given with predefined sentence templates like "it contains __". When the blanks are filled with appropriate content, this input conveys a fairly significant description about the secret word. Last game that Ahn designed is Tagatune [8]. It aims to transform the work of tagging music clips into a game. It works much like ESP Game. But it seems like it could not be that successful mainly because it is difficult to agree on a common word to describe the clip and listening to a sound could take a bit and become boring.

In [9], a method for collecting alternative forms of phrases, namely paraphrases is discussed. For achieving their goal, they develop a web site where people cooperate. The most important component of the system is their partial hinting system. By default, they already have 2-3 paraphrases. But they want to increase this number. This is achieved with partial hints. At the start of the game, no hint is given and users are expected to enter paraphrases of their own. If they are able to guess the already known paraphrases, this contributes to the confidence of that paraphrase. Otherwise, the contribution is stored as a new paraphrase to be guessed by other contributors. This much like resembles social bookmarking sites in which each contribution is accumulated and more submissions of the same contribution reinforces the importance of it. After guessing a paraphrase, if it is unsuccessful, the partial hinting mechanism reveals 33 per cent of the already obtained paraphrases like "this ... help". In [10], five design decisions are introduced. First, it is important to fine tune templates which will collect semantic information (abstract morphological data in our case). Besides fine tuning, it is necessary to provide guidance to users. It is also advisable to break the annotation process into several steps to be able to distribute the work among users. This way multiple users can validate the annotations. Also it would be good to have a way to automatically repair the contributions at least to some extent.

[1] A term coined by Luis von Ahn to refer to the term "CPU cycles"

In [11], it is suggested to have a reward mechanism, which is not only instant rewards after successful annotation but also awards points when another player makes the same annotation at some future time.

A semi-collaborative approach to corpus annotation is described in [12]. But the system simply acts as a data repository that can be accessed simultaneously both online or offline ([13] is also similar in this way). This makes the system miss the collaboration possibility. However, a well thought mechanism is implemented: the contributors are presented with a readily annotated text which is output by a program which accomplishes the task that the collected corpora will help developing programs for. We think this can be further extended to incorporate active learning in the system.

A work by Gülşen Eryiğit [14] describes a standalone (non-web) program which can be used as a tool for dedicated contributors. Relying on specially trained people to annotate the corpus is destined to be slow and costly, despite the increase in speed by using this tool.

In [15], several users can annotate the corpus individually, and later one "consensus user" selects the best annotation. Thus, we think the cooperation aspect of the project is not incorporated by design. Additionally, contribution requires specialized knowledge in the area and no ordinary user can help readily.

As our focus in the paper is to build an unambigously annotated corpus for morphological disambiguation of Turkish, we would like to list some of the current approaches to the problem. A trigram-based statistical model is presented in [16]. In [17], a decision list induction algorithm is introduced for performing morphological disambiguation. There are also several constraint-based methods for disambiguation [18, 19]. Another method employs a perceptron algorithm for morphological disambiguation [20]. We use the tool produced by this study as a morphological parser ranging from preparing the corpus to the online question generation.

3 The Game

We continue with elaborating on the crucial properties which the game must possess. First of all, the game must be playable by ordinary people who are not necessarily educated in the field. This means that we have to find a way to break up the disambiguation process into pieces to be able to tailor the process for non-experts.

At this point, we assume that humans are equipped with a covert ability to sense the correct parse of the word. This ability is learned in the childhood but there is no known way of consistently describing this ability so that it can be programmed to be executed on computers. Thus it seems reasonable to generate all possibilities with a morphological parser and then somehow make the user select the correct parse. One problem here is that these parses cannot be directly understood by a person without knowledge on the subject. Given the facts that humans covertly "know" to separate the good parses form the bad parses and that the raw parses are not sufficiently clear, we find it useful to form questions

acting as an abstraction layer between the user and the raw parses. Thus, we propose to discard bad parses from the set of parses by asking questions of two types; yes/no questions and multi-option questions. These questions must be prepared so that they are automatically generated for any word in the corpus and be clearly understood by the users. By asking this question to a statistically sufficient number of users, we became assured whether the parses that are to be discarded will be discarded or not.

Possibly there will be other questions, because one question will discard only a portion of the set of all possible parses. However, after aggregating the users' answers for these questions, we will have discarded all the bad parses. This means that we have finished disambiguation and left with the correct parse.

In conclusion, our game is capable of generating questions for the words in the corpus automatically. These questions are asked in several stages of both the single and two player game. After aggregating sufficient number of answers, the correct parse of the corpus word is detected.

An additional aspect of the game is that it must be publicly accessible by our target population. To provide this, we chose to host the game on a web site which is accessible at any time of the day and without device restriction. One can access the site by just having the standard equipment which is used to browse the web, namely web browsers. Moreover, we allow people to access our game without formal introduction or qualification tests. This is unlike the previous corpus annotation efforts in which nearly all of them require their contributors to be known and recognized by the people responsible with the process. If we recall that they also usually require the contributors to come to a special office where the work is done, the advantage of our approach is recognized better. In summary, we host the game on a publicly accessible site and allow anyone to join and start the annotation. This in turn makes the potential level of participation (thus work accomplished) much higher than the previous annotation methods. If we take into account that the Internet is maybe the most frequently utilized time killing activity, we can assume this potential to grow even more.

Motivation of the users is another issue which is very closely related with the game design and the site that it is contained. We have two basic notions for building and nourishing motivation.

The first is fun. If the game is fun enough, people will begin to grow an addiction to the game instead of other time spending activities which sometimes can be boring in themselves. To provide the fun element to the game, we introduce a special stage in the game. This stage contains similar elements from Taboo and a famous game in which you try to explain some film title to the audience without speaking. As you might recall, in Taboo, similar to the game about explaining film titles, you are trying to convey a specific concept to the audience without using some words which are prohibited from using -even parts of it. This stage of the game, we call it as the taboo stage for simplicity, is activated only when playing the two player game. One of the users are chosen as the teller and the other as the guesser. The objective of the teller is to give clues about some specific word to the guesser to accomplish her own objective which is to guess the

word as fast as possible. The word that is to be conveyed is actually a word in its sentence context. The sentence is shown to both players. But, obviously, the word in question is concealed from the guesser. The two players enjoy a sense of cooperation while the teller gives clues and the guesser tries word after word. At the same time, they are challenged with a time limit that keeps them alive and attached to the game.

The other aspect of the game which is thought to increase motivation is competition. Naturally, people tend to compete with other people when challenged with a fairly hard problem. The key point here is to design the game so that it is neither too hard nor too easy. We employed several methods for building motivation. The run against the time limit in Stage 2 is itself a competitive factor. In that stage, players compete against the time cooperating with the other player. This forms the basic motivation for the game. Another method is to build motivation by introducing competition based on group membership. This idea is based on the fact that it is known that people form around groups to enjoy group membership advantages. These advantages can vary from just declaring that someone is a member of a prestigious group to gaining benefits for themselves by using the connections among the group. The site which the game is embedded provides users a way to create and join groups as they wish. People can create groups to represent their school, their football team or a way of thinking. People can also do this for completely arbitrary groups. When a group is created, anyone who wants to join is allowed, and as a result the points that are earned by that user are added to the total points of the group. Competition among the groups are thus constituted. We expect to see the total motivation to build up as a result of this competition.

Another dimenson of the competition factor in the game is to focus on individual representation. As it can be guessed, besides group membership, people pay attention to keep their online presences in a state which is desirable by other people. And to do that, people may want to devote a lot of time to earn high points in a game if the result is to be presented to a lot of audience as a highly skilled person. Thus, in order to exploit this behaviour, we present the highest scoring ten users on the home page of the game site. We assume that people will be motivated to get into that list.

3.1 Single Player Game

In single player game mode, the player is first shown a sentence from the corpus. One of the words in the sentence is marked with a distinctive color, namely red. The player is asked a question that is designed to detect a morphological feature of the indicated word. The answer of the player is stored, the player is awarded 50 points, and the game advances. The next stage is actually the same as the previous stage but this time another word from another sentence is selected and displayed along with its context. The game continues until it is ended by the player herself.

The target words are selected so that it is made sure that every type of question gets a statistically significant number of answers. To make the player

answer in a reasonable time, there is a time limit on this stage which was set to two minutes during the experiment.

3.2 Two Player Game

Before starting a two player game, the system matches two users who indicate that they are willing to join a two player game session. After a pair is matched up, they are registered for the same game session. The game session consists of games that are played consequently. The rules of winning a game session is that you have to win all the ten games in a row. If you are not able to win a game in the process, you are not allowed to go to the next game and as a result the game session ends.

We call one of the players as "the teller", the other as "the guesser" throughout a game.

A game of two player mode consists of three stages:

1. the question is asked to the teller
2. the taboo stage
3. the question is asked to the guesser

Stage 1 is basically the same with the single game mode which is explained in Section 3.1. The answer submitted by the player is stored and the player is awarded 50 points. Then, the game advances to the next stage. Meanwhile, the guesser waits for the teller to answer the question while the game displays the same sentence but the target word is concealed. This is to warm up the guesser to Stage 2 and help her to build up some excitement instead of waiting tediously.

In Stage 2 which we call the taboo stage, the same sentence and the indicated word is shown to the teller. But the guesser still does not see the concealed word. The objective of this stage is to operate collaboratively to guess the word as quick as possible. Through an interface which they can communicate simultaneously, the teller tries to give as many clues as possible while the guesser acts upon these clues to guess the target word.

The interface for the teller is different from the interface of the guesser. While the guesser can only utilize a single text box to submit her guesses, the teller's interface contains much more text boxes (see Figure 1). There are a total of nine boxes which the teller can fill with clues. However, each of these boxes differ in the meaning they convey when used. The first box is for clues that are input in free form. While it would be sufficient for the communication between the users, we design the remaining boxes so that each of them reflects another semantic relation between the clue input and the target word itself. We call them clue templates.

The motivation behind these additional text boxes is to gather more fine-grained information about the target word. In fact, we see this is a side effect of the proposed game. A game feature which we add to make the game fun turns out to be helpful for another purpose in the end. This extra information about the word itself possibly can be used for sense tagging. We include the actual

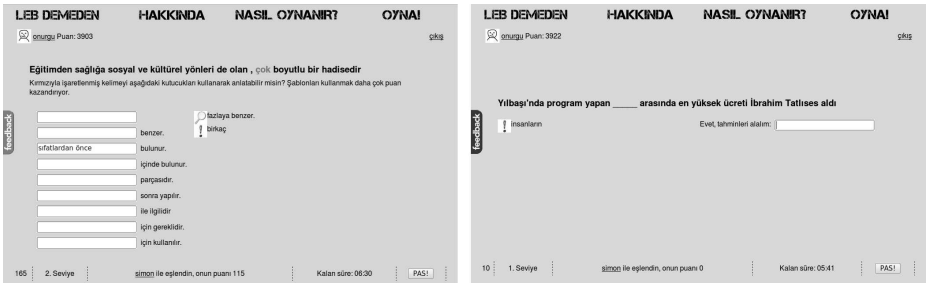

Fig. 1. Teller and Guesser Interfaces (respectively)

decriptive text on two of these clue templates and the meanings associated in Table 1. The points you get is higher if you use the text boxes which correspond to semantic relations. The actual numbers are 5 to 50 points which indicates a factor of ten between the two numbers.

Table 1. Clue Templates

Clue Template	Semantic Relation	Description
____ benzer.	Similarity	Defines a similarity between two objects.
____ bulunur.	LocationOf	Location information.

We had to implement a filter to prevent cheating using these boxes. If we recall the experience obtained from previous work, the participants in these kind of games that offer you fame and some kind of identity representation medium often try to cheat to get those awards more easily (see [4]). The filtering mechanism works like this: First it is checked whether the clue text as a whole can be found in the text of target word, if it is found, the clue is discarded. If it is not, it is checked whether the text of target word can be found in the clue text, if it is found, the clue is discarded, otherwise the clue is accepted. When the clue is discarded, it is not shown to the other user not even partly.

While the interfaces for the teller and the guesser differ generally, there is indeed a widget which is common to both of them. This widget displays the conversation between the teller and the guesser in a sequential manner. As a new guess or clue is submitted, the widget is updated.

We chose a time limit of ten minutes for this stage. This limit is intended to encourage participation in fear of not being able to complete the stage. As you might expect, this stage continues until either the time limit expires or the pair succeeds in guessing the word correctly. Regardless of the situation, we advance to the next stage. However, if they could not guess the target word, the whole game session finishes after the next stage. Each guess from the guesser receives 10 points. Each free text clue is awarded by giving out 5 points. However, if the

clue is submitted using the clue templates, the teller earns 50 points. When the pair successfully guess the target word, they receive 500 points.

In the third and the last stage of this game, the guesser is exposed the same question as the teller in Stage 1. None of the settings differ from Stage 1. Basically, the stage is designed to guarantee obtaining answers from different people for each question. After Stage 3 is finished, the game session goes on with another game if the target word is guessed successfully in Stage 2. If the number of consequent games that were successful reaches ten, we say that the game session finishes successfully and the pair is taken back to the game lounge with a greeting note. As a result of this row of winning games, they are both awarded 5000 points. On the other hand, in case Stage 2 was unsuccessful, the game session is finished and they receive no points.

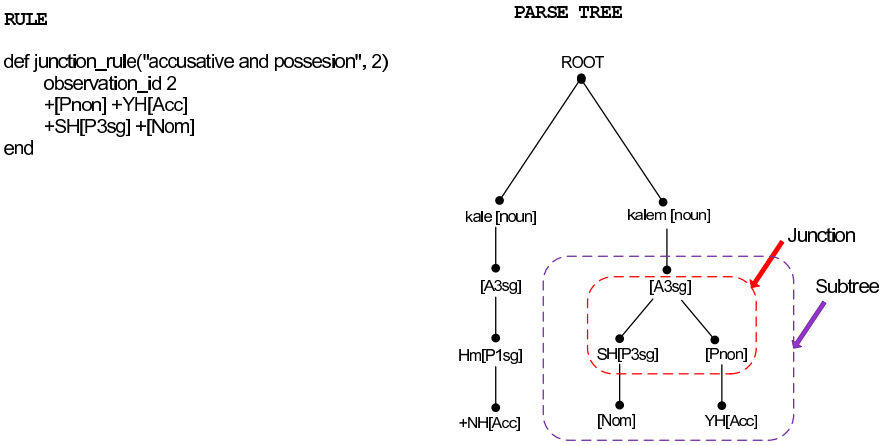

Fig. 2. A Junction Rule and the Corresponding Subtree.

4 Results

The experiment had been done through a game site which is accessible publicly on the web[2]. While it is continuing its operation, we only use the data collected between 29 June 2009 and 9 July 2009, approximately 6000 answers.

According to our experiment plan, we prepared two lists of each having an instance of 74 possible question types and collected about 30 answers for each of them. By doing this, we were able to assess the quality of the questions over two instances, calling the first as Phase 1, and the second as Phase 2. We had to resort to this plan because after speculating on the expected number of visitors, we calculated that it could be infeasible to evaluate our method on the basis of complete disambiguation.

[2] http://lebdemedenleblebi.com

To understand the success criterion of a question, we must first explain the question generation methodology. To generate a question, we first start with enumerating the set of all morphological parses of the word by using a morphological analyzer. We then transform it into a tree. After this transformation, the detection of junction points by observation rules results in abstract objects called observations (see Figure 2). These observations are then matched with question rules. Each matched question rule is applied to the word to generate the unique questions which are tailored solely for determining the correct way to choose in the junction that is represented by the observation. After 30 people answer the question, we agree on the option with most submissions. We verify this agreement answer by checking whether the correct parse reported in the corpus contains the resolution parse tag that is attached to each option.

We calculate the rate of successful questions in Phase 1 as 79.7 per cent. This figure is realized as 71.6 per cent in Phase 2. However, we want to report that a little modification to the definition of a successful question would increase these values to 87.8 per cent and 79.7 per cent. This modification would be to discard the answers of type 'None' or 'I did not understand the question' if they are the highest ones. We observed that this modification increases the rates but in any way we did not change the evaluation method so that to allow an elaboration. When we look at the combined results of these two phases, we see that the percentage of question types that are successful in both of these phases is 63.5 per cent.

There were 400 users registered on the site at the end of the experiment period. A total of 5284 games were played of which 4784 of them was in single player mode. Although the total number of clue templates were utilized only to a certain extent, the users who employed them used 3 templates on average. Experiments show that average time required to answer a question in Stage 1 or 3 took around 36 seconds and the most of the pairs completed Stage 2 well below 60 seconds as can be seen in Figure 3.

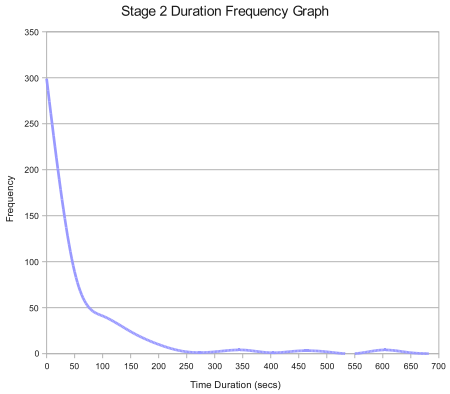

Fig. 3. Stage 2 Duration Frequency Graph

5 Conclusion and Future Work

In this work, a game for morphological annotation of a Turkish corpus is developed. This is the first work that incorporates human computation methods in corpus annotation. The game is meant to be played by two players simultaneously over the Internet. Basically, the annotation is done by collecting answers to questions that are automatically created based on a number of templates prepared manually. In one of the three stages of the two player game mode, one of the players has to describe the target word to the other player trying to collaboratively guess the word as fast as possible. The answers to the questions posed in the other stages are then analyzed statistically and an aggregation of aggreement answers is built which in turn results in a complete morphological disambiguation.

The game is hosted on a publicly accessible web site. The results reported in the paper are compiled from the data obtained between 29 June 2009 and 9 July 2009. The evaluation was done by assesing the performance of all question types over two instances. The reported success rate over the two phases is 63.5%.

As a future work, we see that incorporating an awarding system that can measure the performance of the players and award accordingly can be more facilitating. Also, a method for measuring the difficulty of a question or at least categorizing them by hand would enable us to modify the game so that the levels become harder and harder, thus making the game more challenging. Another important future work is to host the game in a site with a high number of daily visitors to test our method in a real setting and succeed in a complete disambiguation of arbitrary text.

References

1. Marquez, L., Salgado, J.G.: Machine learning and natural language processing (2000)
2. Marcus, M.P., Santorini, B., Marcinkiewicz, M.A.: Building a large annotated corpus of english: The penn treebank. Computational Linguistics **19**(2) (1994) 313–330
3. Yüret, D.: Morphologically tagged corpus (2009)
4. von Ahn, L., Dabbish, L.: Labeling images with a computer game. In: CHI '04: Proceedings of the SIGCHI conference on Human factors in computing systems, New York, NY, USA, ACM (2004) 319–326
5. von Ahn, L., Liu, R., Blum, M.: Peekaboom: a game for locating objects in images. In: CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems, New York, NY, USA, ACM (2006) 55–64
6. von Ahn, L., Ginosar, S., Kedia, M., Liu, R., Blum, M.: Improving accessibility of the web with a computer game. In: CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems, New York, NY, USA, ACM (2006) 79–82
7. von Ahn, L., Kedia, M., Blum, M.: Verbosity: a game for collecting common-sense facts. In: CHI '06: Proceedings of the SIGCHI conference on Human Factors in computing systems, New York, NY, USA, ACM (2006) 75–78

8. Law, E.L.M., von Ahn, L., Dannenberg, R.B., Crawford, M.: Tagatune: A game for music and sound annotation. In: ISMIR 2007: 8th International Conference on Music Information Retrieval. (2007)
9. Chklovski, T.: Collecting paraphrase corpora from volunteer contributors. In: K-CAP '05: Proceedings of the 3rd international conference on Knowledge capture, New York, NY, USA, ACM (2005) 115–120
10. Chklovski, T., Gil, Y.: Improving the design of intelligent acquisition interfaces for collecting world knowledge from web contributors. In: K-CAP '05: Proceedings of the 3rd international conference on Knowledge capture, New York, NY, USA, ACM (2005) 35–42
11. Richardson, M., Domingos, P.: Building large knowledge bases by mass collaboration. In: K-CAP '03: Proceedings of the 2nd international conference on Knowledge capture, New York, NY, USA, ACM (2003) 129–137
12. Bontcheva, C.T., Cunningham, H., Tablan, V., Bontcheva, K., Dimitrov, M., Lab, O.: Language engineering tools for collaborative corpus annotation. In: In Proceedings of Corpus Linguistics 2003, Wiley (2003) 80–87
13. Ma, X., Lee, H., Bird, S., Maeda, K.: Models and tools for collaborative annotation. In: Proceedings of the Third International Conference on Language Resources and Evaluation, Paris: European Language Resources Association. (2002)
14. Eryiğit, G.: ITU treebank annotation tool. In: Proceedings of the Linguistic Annotation Workshop, Prague, Czech Republic, Association for Computational Linguistics (June 2007) 117–120
15. Stührenberg, M., Goecke, D., Diewald, N., Mehler, A., Cramer, I.: Web-based annotation of anaphoric relations and lexical chains. In: Proceedings of the Linguistic Annotation Workshop, Prague, Czech Republic, Association for Computational Linguistics (June 2007) 140–147
16. Hakkani-Tür, D.Z., Oflazer, K., Tür, G.: Statistical morphological disambiguation for agglutinative languages. In: Proceedings of the 18th conference on Computational linguistics, Morristown, NJ, USA, Association for Computational Linguistics (2000) 285–291
17. Yuret, D., Türe, F.: Learning morphological disambiguation rules for turkish. In: HLT-NAACL 06. (June 2006)
18. Oflazer, K., Tur, G.: Combining hand-crafted rules and unsupervised learning in constraint-based morphological disambiguation. In Brill, E., Church, K., eds.: Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Somerset, New Jersey (1996) 69–81
19. Oflazer, K., Tür, G., Tfir, G.: Morphological disambiguation by voting constraints. In: Proceedings of the Thirty-Fifth Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics. (1997) 222–229
20. Sak, H., Güngör, T., Saraçlar, M.: Morphological disambiguation of Turkish text with perceptron algorithm. In: CICLing 2007. Volume LNCS 4394. (2007) 107–118