Introducing Computational Linguistics and NLP to High School Students

Rositsa Dekova
Plovdiv University Paisii Hilendarski
rosdek@uni-plovdiv.bg

Adelina Radeva
Sofia University St.Climent Ohridski
radeva.adelina@gmail.com

Abstract

The paper addresses a possible way of introducing core concepts of Computational Linguistics through problems given at the linguistic contests organized for high school students in Bulgaria and abroad. Following a brief presentation of the foundation and the underlying objective of these contests, we outline some of the types of problems as reflecting the different levels of language processing and the diversity of approaches and tasks to be solved. By presenting the variety of problems given so far through the years, we would like to attract the attention of the academic community to this captivating method through which high school students might be acquainted with the challenges and the main goals of Computational Linguistics (CL) and Natural Language Processing (NLP)[1].

1. Introduction

The Bulgarian linguistic contests for high school students[2] were founded back in the 1980s by Prof. Ruslan Mitkov (Mitkov, 2006a). Called initially "Competitions in Mathematical and Computational Linguistics", these contests targeted highly motivated students, primarily from Mathematical and Language High Schools, with the primary goal of getting students interested in linguistics as a whole and introducing them to some of the core concepts of computational linguistics, thus providing "a promising springboard for future careers in NLP" (Mitkov, 2006a). And they did so, indeed. Many students have acquired their initial knowledge of what computational linguistics is mainly through solving such problems. Some of those students have in turn grown up to be computational linguists and carry on the work in other countries. Most notably, Dragomir Radev – a professor of Computer Science at Yale University (working on natural language processing and information retrieval), author of many problems in computational linguistics (Radev, 2013a, 2013b) given at the National Olympiads in the USA, Canada, Australia, etc. But there are many other Bulgarians, such as Yova Kementchedjhieva (MSc in Cognitive Science Informatics, University of Edinburgh, currently a PhD student in CL, University of Copenhagen), Nikolay Bogoychev (BSc in Artificial Intelligence and CS, University of Edinburgh, PhD student in Informatics at the University of Edinburgh, exploring the application of GPUs in NLP), Dimitar Hristov (BSc in Computer science, University of Southampton, Master's Degree in CL, Sofia University, currently a researcher at the Department of Computational Linguistics, Bulgarian Academy of Sciences), Lyubomir Zlatkov, Pavel Sofroniev, and Stela Ilieva (Bachelor Degree in CL, University of Tubingen), Ivaylo Grozdev (Bachelor Degree in CL, University of Edinburgh), Todor Tchervenkov (Linguistics, Computer Science, Trinity College, Dublin), to mention just a few.

[1] We would like to thank the anonymous reviewers for their valuable remarks and comments.
[2] Throughout the paper, we will also use 'students' and 'undergraduate students' interchangeably to 'high school students'.

Keywords: Teaching Computational Linguistics, Olympiads in Linguistics, NLP

Later, the name of the contests in Bulgaria was shortened to "Competitions in Mathematical Linguistics", but CL oriented problems continued to be included whenever possible.

An important feature of all linguistic problems, designed for these competitions, is that they are created as self-sufficient, i.e. anyone could solve them with no prior theoretical knowledge, therefore allowing students to explore the fundamental concepts and guiding rules on their own, which is far more fascinating than learning theory by heart.

The paper aims at outlining the most general types of NLP tasks and CL applications presented as problems given at different events for undergraduate students such as seminars, summer schools, competitions, national and international Olympiads. Thus, we would like to showcase this alternative method of introducing computational linguistics to high-school students and engage more academics in the field.

2. Types of linguistic problems based on CL applications and NLP tasks

The problems are presented to the students in an accessible and entertaining way. Nevertheless, they outline samples of real NLP concepts, theoretic and practice examples of finite-state transducers, formal grammars and natural language generation, automatic text processing, incl. anaphora resolution, summarization, word sense disambiguation and word sense representation, machine translation, etc. Many linguists have contributed to authoring problems for these contests: Ruslan Mitkov, Dragomir Radev, Ivan Derzhanski, Tom McCoy, Harold Somers, Christiane Fellbaum, Jonathan May, Patrick Littell, Emily Bender, Jonathan Kummerfeld, Tom Payne, Daniel Lovsted, Richard Sproat, Andrea Schalley, Aleka Blackwell, Adam Hesterberg, Ben King, among others.

For the purposes of this article the authors have reviewed and categorized the most frequently given types of CL problems, including problems given at the Bulgarian National Olympiad in Linguistics, North American Computational Linguistics Olympiad (NACLO), established in 2006, Australian Computational and Linguistics Olympiad (OzCLO), established in 2008, All Ireland Linguistics Olympiad (AILO), established in 2009, United Kingdom Linguistics Olympiad (UKLO), established in 2010[3].

2.1. Finite-State Transducers / Finite-State Automata

The problems related to Finite-State Transducers (FST) or Finite-State Automata (FSA) present examples of machines with finite set of states, including initial state and one or more final states. The FST transduces strings of symbols into other strings of symbols. The machines can make successful and unsuccessful transformations using different data – alphabet symbols, numbers, words, etc. The students are given examples of FST and how the transducer works, whereby the task is to repeat a transformation using a set of given data, or to create new transformations.

FST/FSA can be applied on different levels of language processing – phonology, morphology, syntax, or used for coding a text, as shown respectively in 2.1.1, 2.1.2, 2.1.3, and 2.1.4 below.

2.1.1. Phonemic FST and FSA

The concept of the limited system of phonemes in a natural language and therefore the possibility of defining rules for generating words is demonstrated quite well in the problem "aw-TOM-uh-tuh", created by Patrick Littell (NACLO 2008, OzCLO 2008)[4] and also available in Bulgarian (Derzhanski and Velinov, 2012). The problem illustrates an FSA that can distinguish between possible and impossible words in Rotokas[5], which possesses one of the smallest phoneme inventories. The problem represents the FSA as a board game (see Fig. 1), allowing students to understand how to generate possible words with a given set of phonemes and rules for their combination.

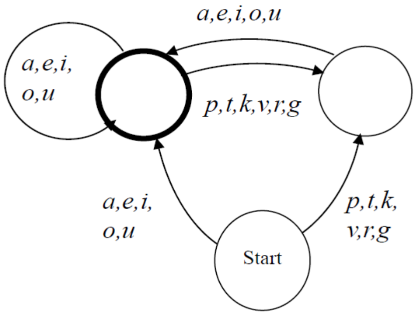

Figure 1: FSA for possible words in Rotokas

[3] Past problems and practice samples could be found on most of the National Olympiads web-pages (cf. References).

[4] Available online at: `http://www.nacloweb.org/resources/problems/2008/N2008-I.pdf`

[5] An isolated language, spoken on the island of Bougainville, east of Papua New Guinea

Another problem demonstrating phonemic transformations with an FST is created by Tom Payne (NACLO Sample Practice Problems)[6] and named "Computational Machines". It shows a diagram of an FST that transforms the English word "cat" into the English word "dog" in three steps. It also provides an example of a machine that allows for an infinite number of inputs. Thus, the problem urges students to differentiate between the possible and the desired outputs. The task that should be accomplished by the students is to create a similar diagram that will transform "Tom Cruise" into "Ali Landry" using four circles or less.

2.1.2. Morphemic FST and FSA

The problem "Transition(al) numbers", created by Harold Somers (NACLO 2017)[7], illustrates the use of a morphemic FSA (presented as a "transitional network") illustrating the set of rules generating the English numerals smaller than a hundred. The students also learn the concept of "overgeneration" and are asked to correct some of the rules or to create new ones in order to "fix" the network.

2.1.3. Syntactic FST and FSA

Finite-State Automata can also be used to illustrate simplified sentence generation. Three similar problems (Mitkov, 1989: 79-80) offer examples of such FSA, where every arc connecting two states represents a different word, and the generated sentences can be grammatically correct or incorrect. The students' tasks are to propose an FSA using the same words but with different states and directions of the arcs, so that all the generated sentences are grammatically correct (compare the FTA presented to the students (Fig.2) with the FTA expected to be produced by the students as a solution to the problem in Fig. 3 below).

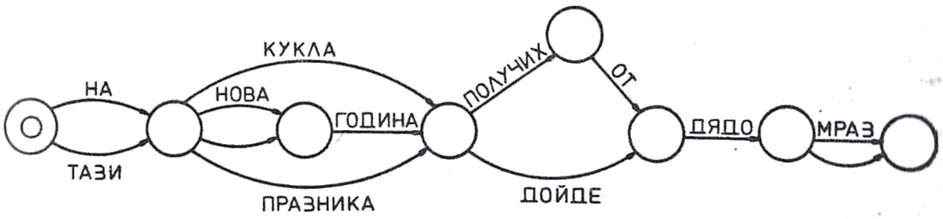

Figure 2: Sample syntactic FSA[8]

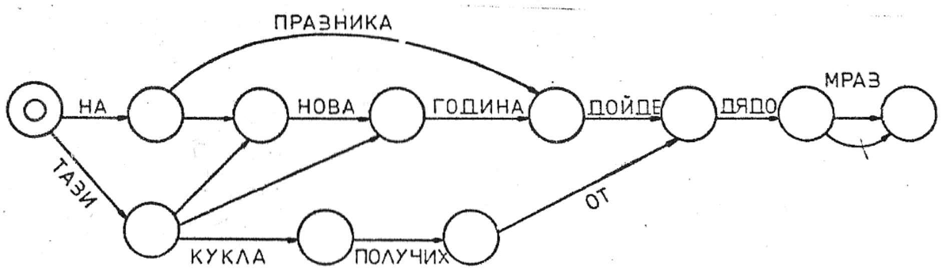

Figure 3: Problem Solution

2.1.4. FST and FSA for encoding text

Being an effective means in various types of automatic treatment of natural languages, Finite-State Transducers can also be used for coding and decoding texts. The problem "Finite-State Transducers" created by Richard Sproat (NACLO Sample Practice Problems)[9] illustrates how the input alphabet describes a recognizable pattern that is transformed into the output alphabet, and how that can be applied for a text. The students' task is to decode a sample of text, a simple kind of deciphering, using the provided output data, and a diagram of the FST used in the initial ciphering.

[6] Available online at: `http://www.nacloweb.org/resources/problems/sample/FST-4.pdf`

[7] Available online at: `http://www.nacloweb.org/resources/problems/2017/N2017-F.pdf`

[8] Fig. 2 is used in a problem by I. Nenkova (Mitkov, 1989: 79-80) and Fig. 3 is given as a solution to the problem (ibid. 151).

[9] Available online at: `http://www.nacloweb.org/resources/problems/sample/FST1.pdf`

2.2. Formal grammars and natural language generation

Problems presenting formal rules for generating words or sentences are another great illustration of introducing CL to students. The “Grammar Rules!” problem, created by Patrick Littell and Andrea Schaley (NACLO 2013, AILO 2013, OzCLO 2013)[10] sets forth the notion of a Context free grammar presented as a set of phrase structure rules of the type (S → NP VP, NP → N, etc.). The students are then supplied with a number of sentences and their task is to select the sentences that are well formed according to this CFG, i.e. which can be generated by the given rules. Also, they must find the instances of overgenerated sentences, i.e. the ones that are well formed by the CFG rules but not grammatical (according to the rules of English). Finally, students must also detect the one rule that is redundant.

Another problem also employing a CFG is called “Fan Fiction” by Ben King (NACLO 2016, OzCLO 2016)[11]. The story revolves around a fan-fiction writing robot who can use a few different methods for generating sentences, such as n-gram methods (unigrams, bigrams and trigrams) and a CFG. All the methods are illustrated by simple examples. The students are then provided with a collection of sentences, only three of which are real (written by a human author) and the rest are generated using one of the methods described. The students are to detect which sentences are real and which of the other methods has been used for each of the sentences generated by the bot.

Josh Falk introduces the students to the Horn clause notation (for ex. S(xy) :- N(x), V(y).) in his problem “A Matter of Horn Clauses” (NACLO 2016)[12], where the students are supposed to use the notation to describe English and Swiss German sentences.

There are also other problems which apply formal rules for generating words, instead of sentences. Some examples of such problems are: “Text-o-matic” by Daniel Lovsted (NACLO 2017)[13], which presents rules for generating the paradigm of French numerals; “Minimum Spelling Trees” by John de Nero (NACLO 2015)[14], which involves encoding of German noun forms (generating the paradigm of a word); and “Lexicondensed” by Tom McCoy (NACLO 2014)[15], which introduces formal lexicons for the task of creating Spelling Change Rules for generating a list of adjectival forms of country names.

2.3. Automatic text processing

2.3.1. Anaphora resolution

Anaphora resolution, on the one hand, still presents a challenge for NLP, but on the other hand, it can be transformed into a challenging problem for undergraduate students.

One of the oldest problems featuring this phenomenon was created by Ruslan Mitkov (Mitkov, 2006b: Sample Problem 5) and it was given at the 10th National Competition on Mathematical and Computational Linguistics (Bulgaria 1993). It illustrates a situation where all noun phrases (NPs) preceding the pronoun are regarded as potential antecedents, and the students’ task is to propose filters, which will allow the computer to define the correct antecedents. At the end of the problem, the author provides an additional point of view, asking the students whether this is possible only when the pronoun follows the NP (thus pointing towards the existence of cataphora without explicitly defining the notion).

The problem “Help my camera!”, created by Emily Bender (NACLO 2009, OzCLO 2009)[16] and adapted for Bulgarian students by Ivan Derzhanski (Derzhanski and Velinov, 2012), presents three dialogues between a human and a computer, as the examples below:

Human: *There's this restaurant on Bancroft that's supposed to be really good that I heard about from my mother. Can you help me find it?*
Computer: *Where did you last see your mother?*

[10] Available online at: `http://www.nacloweb.org/resources/problems/2013/N2013-F.pdf`

[11] Available online at: `http://www.nacloweb.org/resources/problems/2016/N2016-H.pdf`

[12] Available online at: `http://www.nacloweb.org/resources/problems/2016/N2016-P.pdf`

[13] Available online at: `http://nacloweb.org/resources/problems/2017/N2017-R.pdf`

[14] Available online at: `http://nacloweb.org/resources/problems/2015/N2015-M.pdf`

[15] Available online at: `http://nacloweb.org/resources/problems/2014/N2014-J.pdf`

[16] Available online at: `http://www.nacloweb.org/resources/problems/2009/N2009-F.pdf`

Human: *My parents are having a hard time choosing a new digital camera. How can I help them?*
Computer: *I'm afraid I don't understand your question. People don't help cameras.*

The computer misunderstands the human because it is misinterpreting certain words or phrases. The students' tasks are to explain why the computer cannot understand the human properly, and to propose better rules for the computer to interpret the anaphoric words and phrases. The problem shows the difference between human and machine way of understanding such relations.

The "Common Sense" problem by Jordan Ho (NACLO 2017)[17] again provides a classic example of how anaphora relates to its antecedent. Students are given examples of anaphoric references, and their task is to relate the pronouns to the correct antecedent (all antecedents are words invented by the author).

2.3.2. Word sense disambiguation and word sense representation

As mentioned above, one of the main advantages of using linguistic problems for introducing CL and NLP tasks to high-school students is that these problems are often quite intriguing and entertaining. Sometimes, this allows for rather complex theoretical issues to be introduced to the students in a simple and enjoyable way. For example, Emily Bender's problem "The Old Man the Boats" (NACLO 2015)[18] presents syntactic ambiguities with a sense of humour. The problem reviews a number of sentences (also known as garden-path sentences) each containing a local ambiguity to be solved, as in the examples:

The old train the young.
I convinced her children to do their homework.
The man who whistles tunes pianos.

After explaining concisely the nature of POS ambiguities resulting in different sentence structures, it requires the students to parse the sentences, to define their local ambiguity point and to provide a new ending after that point so that the other reading of the ambiguous word surfaces.

The problem ″Kings, Queens and Counts″ by Tom McCoy (NACLO 2016) introduces a method of automatically representing word meaning (as shaded graphs) based on the count of its collocations. The students are given a number of diagrams with the most common collocations defining a word based on a sample text, and the meaning of the word itself. Firstly, the students are asked to shade the graph representation of another word from the same sample text. Then their task is complicated – to match 11 mystery words to their definitions using only the information from the representations of 33 words (incl. the mystery words), obtained from a different sample text.

2.3.3. Word categorization

There are a lot of problems which ask students to categorize (unknown) words based on the context. Created by Dragomir Radev, Christiane Fellbaum and Jonathan May, the problem called "Zoink!" (NACLO 2015, UKLO 2015, AILO 2015)[19] engages the students in helping the administrators of the website Zoink! to determine whether their reviews are written by bots or real people. To do that the administrators must write filtering software categorizing words into three groups based on a corpus (set of snippets from reviews written by humans). The students' task is to categorize the new data (another set of snippets) into the same three groups: written by human (correct), written by bots (wrong), and undefined (maybe). The problem explores the linguistic phenomenon of the specific structure of arranging multiple adjectives that describe different degrees of a quality (the correct 'good but not great' vs. the ungrammatical 'furious but not angry', and the undefined 'furious but not good').

"Gelda's House of Gelbergarg" by Patrick Littell (NACLO 2010, UKLO 2010, OzCLO 2010)[20], presents a similar model of categorizing unknown words into a) individual, discrete food items; b) liquids, undifferentiated masses or masses of uncountably small things; and c) containers or

[17] Available online at: `http://www.nacloweb.org/resources/problems/2017/N2017-O.pdf`

[18] Available online at: `http://nacloweb.org/resources/problems/2015/N2015-P.pdf`

[19] Available online at: `http://www.nacloweb.org/resources/problems/2015/N2015-G.pdf`

[20] Available online at: `http://nacloweb.org/resources/problems/2010/A.pdf`

measurements. Again, decisions must be made based on context, presented in the form of customers' reviews, as in the examples below:

> *A hidden gem in Lower Uptown! Get the färsel-försel with gorse-weebel and you'll have a happy stomach for a week. And top it off with a flebba of sweet-bolger while you're at it!*
>
> *The portions at this place are just too big! I'd rather have half the portions at a lower price – they just bring out too many göngerplose and too much meembel for me.*

2.3.4. Summarization

Automatic summarization aims to create an abstract with the major points of the original document. In a problem given at the 11th National Competition on Mathematical and Computational Linguistics (Bulgaria 1994), Mitkov (2006b, Sample problem 6) displays a case where a computer program must summarize a given document without understanding it, using only a set of predefined "selection" and "rejection" rules. The students' task is to propose three rules of each kind. The rules should not include any morphological, syntactic, semantic or pragmatic analysis.

Another problem involving automated summarization is "Summer Eyes" by Dragomir Radev and Adam Hesterberg (NACLO 2009)[21]. The students are presented with the inputs and outputs of an extractive summarizer and the scores assigned by the summarizer for each sentence according to some criteria which mark it as a good summary sentence. The students are then asked to guess the criteria and rescore the sentences after a change in the story. The criteria which the students should discover include the primacy or recency of a sentence, the presence of named entities and words from the title, and choice between past- vs. present-tense verbs.

2.4. Machine translation

Even if originating in the middle of the 20th century, the idea of machine translation as the process of translation of one natural language to another, using computational software, still captivates our efforts as researchers and presents many challenges. Therefore, there is a variety of concerns to be addressed in the field. The linguistic problems related to MT evolve with each new approach. Some of the earliest problems regarded rule-based MT (Mitkov 1989: 71-75), while the ones that are more recent relate to different aspects of statistical MT. For example, "Running on MT" by Harold Sommers (NACLO 2011, UKLO 2011)[22] points out the problem of word sense disambiguation for the purposes of machine translation. To simulate the effect of automatically selecting a word sense, a number of individual words from an ordinary English text were replaced with alternative words which share a meaning with the original word, but which were not correct in this context, as presented in the sample below:

> *Annie Jones sat* ^{cross} ~~*angry*~~*-legged on her Uncle John's facade porch; her favorite rag doll clutched under one supply. The deceased afternoon sun polished through the departs of the giant oak tree, casting its flickering ignite on the cabin. This entranced the child and she sat with her confront changed upward, as if hypnotized. A stabilize hum of conversation flowed from inside of the cabin.*

As MT could be based on a number of different methods, applying a variety of approaches, and include numerous subtasks, problems may vary a lot. Future linguistic problems might as well include various notions of Machine Learning and analysis of simple inputs and outputs of Neural Networks, for example, as in fact does one of the latest NACLO problems "Nothing But Net(works)" created by Tom McCoy (NACLO 2017)[23].

Despite the impressive advance of new technologies, however, we believe that CL problems should remain self-sufficient in nature and technologically independent to reach out to students regardless of their prior knowledge and status.

[21] Available online at: `http://www.nacloweb.org/resources/problems/2009/N2009-E.pdf`

[22] Available online at: `http://www.naclo.cs.cmu.edu/problems2011/A.pdf`

[23] Available online at: `http://www.nacloweb.org/resources/problems/2017/N2017-H.pdf`

3. Conclusions

Outlining different problems designed for the needs of linguistic contests, we attempted to show a possible enthralling, yet effective way of introducing CL concepts and NLP tasks to high school students. Besides the examples presented in the paper, there are other NLP tasks which may (and in fact are) presented in a problem: spelling correction, optical character recognition and handwriting recognition, expansion of abbreviations, named entity classification, sentence boundary identification, etc.

A detailed statistical survey of the problems by year and type would not be very informative as the nature of the problems presupposes their relative uniqueness. Thus, a specific CL topic could be used just once unless a completely new scenario is suggested using the same underlying CL task or method.

Then a broader spectrum overview reveals the general tendencies – the more used a method is for solving different CL or NLP tasks, the more likely it is to appear in a new problem; and the other way round – the variety of methods employed to complete a specific CL or NLP task or application correlates directly to the variety of scenarios that can be created.

We believe that throwing some light on the issue will facilitate the involvement of more academics and researchers in this undertaking and will encourage their interest in creating new problems exploring recent methods used in CL and NLP.

References

All Ireland Linguistics Olympiad (AILO). `https://ailo.adaptcentre.ie/`

Australian Computational and Linguistics Olympiad (OzCLO). `http://ozclo.org.au/`

Derzhanski, I. Velinov, A. (2010). *Lingvistichna mozaika*. Prosveta Publishing House.

Derzhanski, I. Velinov, A. (2012). *Lingvistichen kaleidoskop.* Prosveta Publishing House.

Mitkov, R. (1989). *V pomosht na izvanklasnata rabota po matematicheska i kompyutyurna lingvistika*. Natsionalen tsentar za uchenichesko i nauchno tvorchestvo. Sofia.

Mitkov, R. (2006a). *Brief Outline of the School Activities in Mathematical and Computational Linguistics in Bulgaria in the 1980s and 1990s.* `http://pers-www.wlv.ac.uk/~le1825/ena/outline.pdf`

Mitkov, R. (2006b) *Sample problems offered at Mathematical and Computational Linguistics Competitions in Bulgaria.*`http://pers-www.wlv.ac.uk/~le1825/ena/sample.pdf`

North American Computational Linguistics Olympiad (NACLO). `http://nacloweb.org/`

Radev, D. (2013a). *Puzzles in Logic, Languages and Computation: The Red Book (Recreational Linguistics 1).* Springer.

Radev, D. (2013b). *Puzzles in Logic, Languages and Computation: The Green Book (Recreational Linguistics 2).* Springer.

United Kingdom Linguistics Olympiad (UKLO). `http://www.uklo.org/`