TEACHING COMPUTATIONAL LINGUISTICS: CHALLENGES AND TARGET AUDIENCES

Raquel Amaro[1]

Abstract *— Depending on the standpoint, Computational Linguistics can be defined as a subfield of Computer Science dedicated to the processing of specific data – natural language data – or as a subfield of Linguistics concerned with formal modeling of linguistic knowledge for computation purposes. These two perspectives reflect the two main paths to this interdisciplinary field, but also the challenges posed to its teaching. Namely, focusing on, and mastering, logic reasoning and formal models, for Language and Humanities students, and acknowledging and dealing with irregularity, variation and idiosyncrasy, for Computer Science, Engineering and Technology students. This paper discusses the major obstacles and handicaps that seem to stand in the way of teaching/learning Computational Linguistics, an area with high visibility, appeal and applicability potential, aiming at raising attention to some simple but usually overseen aspects that may improve teaching/learning results.*

Index Terms — Computational Linguistics, Natural Language Processing, target audiences, teaching/learning.

INTRODUCTION

Computational Linguistics is an interdisciplinary discipline that combines linguistic and computer science knowledge. [1] defines it as the study of computational systems aiming at understanding and generating natural language, and focusing on capturing the power of human languages. [2] uses a similar definition but from a different standpoint, stating that Computational Linguistics is the scientific study of natural language under a computational perspective, and so computational linguists have as main goal to develop and provide computational models to account for several types of linguistic phenomena.

As recognized by the Association of Computational Linguistics, "work in Computational Linguistics is in some cases motivated from a scientific perspective in that one is trying to provide a computational explanation for a particular linguistic or psycholinguistic phenomenon and in other cases the motivation may be more purely technological in that one wants to provide a working component of a speech or natural language system." ([3]), reflecting, on the one hand, the interdisciplinary nature of the field, and on the other, the two main academic paths that lead to working in it.

So, Computational Linguistics consists of the use of linguistic theories and computational techniques to tackle problems concerning natural language processing (NLP), with many engineering applications, but it is also the language science that pays special attention to the particular difficulties related to the processing complexity of the human cognitive architecture.

The two main academic paths that lead to the field of Computational Linguistics are studies in Computer Science, via NLP (e.g. Interface systems, knowledge extraction and summarization), Machine Learning (e.g. Machine Translation) and Artificial Intelligence, when dealing with natural language; and Linguistics, via formal models of language (syntactic, semantic, phonological). As expected, these paths can be substantially different and implicate substantially different basic knowledge and skills.

This paper discusses some major obstacles and handicaps that seem to stand in the way of teaching/learning Computational Linguistics and that are directly related to the twofold path leading to it, putting forth some strategies that may improve teaching and learning results.

The paper is organized as follows: the first two sections are dedicated to presenting and exploring each Computational Linguistics teaching/learning environment, i.e., Computer Science, Engineering and Technology courses and Language and Humanities courses, respectively. Each section discusses the relevant concepts, as well as the main challenges and possible strategies to overcome them for each academic path. The fourth section is dedicated to industry requirements concerning Computational Linguistics tasks and professionals, and the last section presents some final remarks.

FROM NATURAL LANGUAGE PROCESSING TO LINGUISTICS

The first relevant notion to discuss when it comes to understanding the background – and related challenges – of Computer Science, Engineering and Technology students is the conceptual difference between *computation* and *processing*.

Computation can be defined as a Math-based concept that, in its primary sense, consists in an algorithmic process, i.e., a process that generates correct results through the performance of an effective procedure. An effective procedure can be defined as an explicit and usually ordered set of rules that assures the production of the correct result

[1] Raquel Amaro, Assistant Professor, Faculty of Social Sciences and Humanities, Universidade Nova de Lisboa - Portugal, raquelamaro@fcsh.unl.pt

DOI 10.14684/WCCSETE.2.2016.38-42

for each relevant input ([4]). The notion of computation enters Cognitive Sciences (and later Computer Sciences) due to logicians and Alan Turing in the 30s, when it was used to characterize brain activities. In this context, a machine is a *computer* because it computes, from what follows that Computational Linguistics is not *computational* because it uses or concerns computers, but rather because it concerns *computation* ([5]).

Processing, on the other hand, is a concept from Systems Engineering and Communications Engineering (from the 40s) that consists in the manipulation of individual signals to transmit information ([6]). The base idea is that organisms (and automata) receive and transmit information within the system, and between the system and the environment. So, information processing is one way of understanding how organisms are aware of what happens around them, allowing them to respond appropriately, and of trying to simulate their behavior. An information processing system can comprehend information coding into a signal, signal transmission and information decoding from the transmitted signal, and usually does not concern assigning meaning to the message ([7]).

This distinction is relevant for the issue at hand because it allows us to understand why NLP so often includes fully automated work or language-independent applications, i.e., with no linguistic knowledge, and why students coming from this path can have difficulties dealing with Computational Linguistics.

Challenges and strategies (I)

BA and MA curricula for Computer Science, Engineering and Technology courses often consider NLP or Computational Linguistics as a specific class, covering specific programming languages or strategies (Prolog, Context Free Grammars) or common applications in these fields, such as character strings treatment, Part-of-Speech tagging, *corpora* search and concordances, interface systems ([8]). However, the myriad of issues and phenomena that are addressed in Computational Linguistics cannot easily be taught/learnt in one semester. Moreover, Computational Linguistics, in fact, cannot be dissociated from its theoretically-oriented counterparts: for instance, syntactic or morphologic computational rules are defined based on already studied and typically well-established syntactic and morphologic knowledge that emerges from Linguistics subfields of Syntax and Morphology. In this way, computational linguists are better prepared with a more comprehensive and theoretical background on core areas of Linguistics such as Phonetics, Morphology, Lexicology, Syntax, Semantics, Pragmatics, even if not all.

One might argue that this perspective would transform a Computer Science course with some focus on NLP on a Computational Linguistics course, with a strong base on Linguistics, which is not our aim. Nonetheless, one of the greatest challenges for Computer Science, Engineering and Technologies students is to look at and consider theoretical Linguistics knowledge as extremely valuable, given it greatly reflects the regularities and systematicity of natural languages ([9]), on the one hand, and often motivates and prompts new approaches further, on the other. For instance, from the Linguistics point of view, an NLP statistic approach to meaning that proposes to look at contexts (i.e. neighboring words of a given word) vertically (that is, at all contexts in a given *corpus* in which a given word appears) as an innovative approach is quite misplaced, given that the well-known linguistic hypothesis, Harris distributional hypothesis, – stating that words occurring in the same contexts tend to have similar meanings – was proposed more than half a century ago ([10]).

For Computer Science, Engineering and Technology students, statistically relevant facts are usually seen as of great value – and often sustain machine-learning approaches. Linguistic knowledge, even if loosely and informally described, is just that – statistically relevant facts –, for a given language or for many/all. E.g., describing Portuguese as a S(ubject) V(erb) O(bject) language is the same as saying that the most common or neutral Portuguese sentences are composed of a noun phrase that precedes the verb and that establishes agreement conditions with it (number and person), a verb that agrees with and selects that noun phrase and that selects and is followed by another noun phrase with the syntactic function of object (internal argument that together with the verb states something about the subject). This is a statistically relevant fact that does not exclude other configurations.

Also, students should realize that all speakers have all this implicit linguistic knowledge that, in spite of being mostly intuitive, is responsible for all kinds of insights and judgments that are essential to understand, describe and represent language phenomena and, thus, quite helpful for NLP tasks.

This assumption is behind some international initiatives such as the International Linguistics Olympiad – one of 12 International Science Olympiads for secondary school students annually held since 2003, in which teams from around the world gather and test their minds against the world's toughest puzzles in languages and Linguistics, no prior knowledge of Linguistics or languages required [11] – or the North American Computational Linguistics Olympiad – "a contest in which high-school students solve linguistic puzzles […] learning about the diversity and consistency of language, while exercising logic skills. No prior knowledge of Linguistics or second languages is necessary. Professionals in Linguistics, Computational Linguistics and Language Technologies use dozens of languages to create engaging problems that represent cutting edge issues in their fields" [12].

Although usually directed at young students, linguistic puzzles are in fact quite efficient when it comes to put students in contact with their intuitions about languages, while showing them data driven methodologies and the

nature, consistency and stability of linguistic knowledge. These exercises also show the significance of idiosyncrasies and language-dependent features (diversity), and the importance of evaluating and rating errors (that is, that there are errors humans tolerate and errors that humans never accept) (linguistic levels and analysis).

FROM LINGUISTICS TO COMPUTATIONAL MODELS AND TASKS

Language and Humanities students usually come from a quite different starting point: they are quite knowledgeable about core subfields of Linguistics (Syntax, Phonetics, Phonology, Semantics), but react poorly to formal systems and representations. This general difficulty can be firstly related to the different main goals and tasks of theoretical Linguistics and Computational Linguistics and NLP: theoretical Linguistics aims at describing linguistic knowledge and explaining how it works, typically with strong theory-based approaches; Computational Linguistics and NLP are more concerned with making this description and representation functional for a given specific task, using the theories better suited or available.

Although seeming a very natural environment, multi-theoretical disciplines are not that usual: teachers tend to present and use the theory in which they work, or with which they are more confortable with, in a focused and deep approach, neglecting that sometimes it is as important that students learn to distinguish knowledge and facts from the theories and models of representation for these knowledge and facts.

As it happens with the path previously described, Language and Humanities MA and BA courses tend to have a single Computational Linguistics discipline, this time focused on 'alternative' models of Syntax – Head-driven Phrase Structure Grammar (HPSG), Lexical Function Grammar (LFG), dependency grammars ([8]) – that rely heavily on formalization. Once again, a single semester is not enough to cover all different subareas, goals and tasks encompassed in Computational Linguistics, along with some basic knowledge in Logic and Programming that usually allows for an easier and more solid learning experience.

Challenges and strategies (II)

For Language and Humanities students, the main challenge in learning Computational Linguistics seems to be the 'aversion' to formal representations. This can be easily explained given that formal models and representations usually use Logic, Math and Computer Science symbols and notions, such as predicates, quantifiers, Boolean operators, set theory concepts and notation, lists, and so on, and Language and Humanities students often do not study Math since middle school. So, an extra effort is required to learn the necessary concepts and metalanguages. As mentioned before, the formal character of theories and models is not the only handicap when dealing with Computational Linguistics theories. The lack of theoretical diversity (i.e., the notion that there might be several alternative theories and models for a given phenomenon, with different strong points and weaknesses) often stands in the way of students. For instance, students are often convinced that the Chomskyan notion and representation of syntactic phrases (noun phrase, verb phrase, prepositional phrase, etc.) is universal and is equal to reality. And this is a quite serious barrier to overcome when faced with a head-modifier rule in the format of attribute-value matrix of HPSG, for instance, although many basic assumptions of this framework are shared with and/or come from Chomsky's generative grammar ([13]).

One strategy to overcome these difficulties is making students using available (and quite simple) tools for implementing linguistic knowledge ([14], [15], [16]). For instance, [14] and [15] provide simple Context-Free Grammars builders and parsers that allow students to build and test grammars such as

S → NP VP .
NP → N .
VP → V NP .
N → dogs .
N → cats .
V → hunt .
V → fear .

that produce the following language (set of sentences):

dogs hunt dogs
dogs hunt cats
dogs fear dogs
dogs fear cats
cats hunt dogs
cats hunt cats
cats fear dogs
cats fear cats

(obtained from [14]).

The use of these available and easily accessible tools and applications is, in fact, a quite simple strategy. But it allows students to grasp and exercise several important notions and skills, such as:

i) there are several adequate ways of reaching the same result (choosing one over the other requires taking into account the goals we are aiming at);

ii) 'traditional' linguistic knowledge can be easily translated into formal and rigid representations;

iii) (computational) linguistic theories are complex because they have to (simple and too rigid formalisms and theories do not explain or cannot reproduce natural languages);

iv) Computational Linguistics and NLP tasks are often concerned with specific and modular goals (i.e., we do not have to account for everything all at once).

ON INDUSTRY REQUIREMENTS

All the issues discussed above are relevant when it comes to industry requirements. Given its range of applications, Computational Linguistics is an area with high applicability and employment potential. The table below presents a summary of areas, tools and applications of Computational Linguistics.

TABLE 1
COMPUTATIONAL LINGUISTICS APPLICABILITY

Area	Tasks/modules	Tools & applications
shallow processing	• word segmentation • hyphenation • lemmatization	optical character recognition; documentation management; speller
↓		
deep processing	• lexical analysis • morphological analysis • syntactic analysis • semantic analysis	syntactic and style checkers; information extraction and retrieval; summarization; machine translation; natural language interface; natural language recognition and generation
↓		
artificial intelligence	• pragmatic analysis • world knowledge • inference and reasoning • learning	natural language interpretation and understanding

Given its nature, industry is goal-oriented, usually focused on solid and tangible results for specific products. But in Computational Linguistics, these may also imply the theoretical knowledge to sustain adequate options.

Consider, for instance, the development of a syntactic checker. This may involve, first and foremost, the analysis, description and understanding of common writing human errors – it would not be useful to detect and correct an error that occurs only once in 200.000 pages. This first task requires only knowledge in Linguistics, and typically language-dependent. For a second task, however, it is necessary to understand the final application/tool to be developed and the available resources the computational linguists will be constrained to work with – will there be a complete lexicon with morphological, syntactic and semantic information or a simple word list? This second task asks for a more interdisciplinary knowledge, merging linguistic and computer science skills and strategies. And, finally, a third task could consist of writing a program that includes a parser (syntactic analysis), identifies the errors and proposes the proper corrections. This work concerns mainly, and naturally, programming skills.

Computational linguists should, in fact, respond to all these challenges, but given the two paths, balanced teams are often composed of professionals from both areas: computer scientists and linguists specialized in NLP or Computational Linguistics. As it would be expected, industry requirements for computational linguists concern:

- goal-oriented approaches (i.e., people that easily accept that what is necessary is to solve a given problem and not all related issues);
- solid basic knowledge (people with solid theoretical background and able to recognize and recommend reliable sources – authors, grammars, lexical resources, *corpora*);
- flexibility and adaptability, more than specific skills in specific programming languages (more often than not, professionals are required to learn and work with programming languages and environments developed specifically by the industry).

FINAL REMARKS

The challenges and target audiences of Computational Linguistics teaching reflect the interdisciplinary nature of this field but also professional requirements of the industry. So, and as final remarks for this discussion, it might be relevant to list some aspects presented so far and raise attention to them.

- Considering teachers with academic backgrounds different from that of the target audience (i.e., Linguistics background for Computer Science, Engineering and Technology students; Computer Science background for Language and Humanities students). This would allow for compensating and strengthening students skills in Linguistics and Logic and Programming background, respectively.
- Using different approaches directed at the different target audiences: analysis of linguistic data and exploitation of implicit linguistic knowledge for Computer Science, Engineering and Technology students, to show them the usefulness of theoretical Linguistics and the relevance of idiosyncrasies and diversity in natural languages for NLP tasks; hands-on exercises with available tools for implementing specific theories and models for Language and Humanities students, to show them the importance of formalization and of the use of formal metalanguages.
- Exposing students to several theories and models, training their flexibility, adaptability and necessary distinction between knowledge, explanations and goals.

Teaching/learning Computational Linguistics may be improved by simple strategies like these, which acknowledge and respect the interdisciplinary nature of this field and the differences of the target audiences that are drawn to it.

REFERENCES

[1] Grisham, R., *Computational Linguistics: An Introduction*, Cambridge University Press, 1986.

[2] Söhn, J.-P., *Introduction to Computational Linguistics*, Tuebingen, 2007.

[3] In http://www.aclweb.org/portal/what-is-cl (last accessed 16/07/2016)

[4] Horswill, I. "What is Computation?", in *XRDS: Crossroads, The ACM Magazine for Students - The Legacy of Alan Turing: Pushing the Boundaries of Computation*, vol. 18 Issue 3, pp. 8-14, Spring, 2012.

[5] Law, E. , "Defining (Human) Computation", in *CHI 2011: Workshop on Crowdsourcing and Human Computation*, 2011.

[6] Ralston, A., *Encyclopedia of computer science*, Nature Pub. Group, 2000.

[7] Denning, P. J. & Bell, T. "The Information Paradox", in *American Scientist*, vol. 100, no. 6., pp. 470–477, 2012.

[8] Bolshakov, I. & Gelbuk, A., *Computational Linguistics: Models, Resources, Applications*, Mexico: IPN, UNAM, FCE, 2004.

[9] Johnson, K., "On the systematicity of language and thought", in *Journal of Philosophy*, vol. 101, no. 3, pp. 111–140, 2004.

[10] Harris, Z., "Distributional structure", in Word, vol. 10, no. 23, pp. 146–162, 1954.

[11] In http://www.ioling.org (last accessed 16/07/2016).

[12] In http://www.nacloweb.org (last accessed 16/07/2016).

[13] Sag, I. & Wasow, T., *Syntactic Theory – A Formal Introduction*, Standford: CSLI Publications, 2006.

[14] CFG generator, http://mdaines.github.io/grammophone/ (last accessed 26/07/2016).

[15] CFG builder and parser, http://languagelink.let.uu.nl/~lion/?s=Playgrounds/CFG_Playground&lang=en (last accessed 26/07/2016)

[16] Regular Expression builder & tester, http://regexr.com (last accessed 26/07/2016).