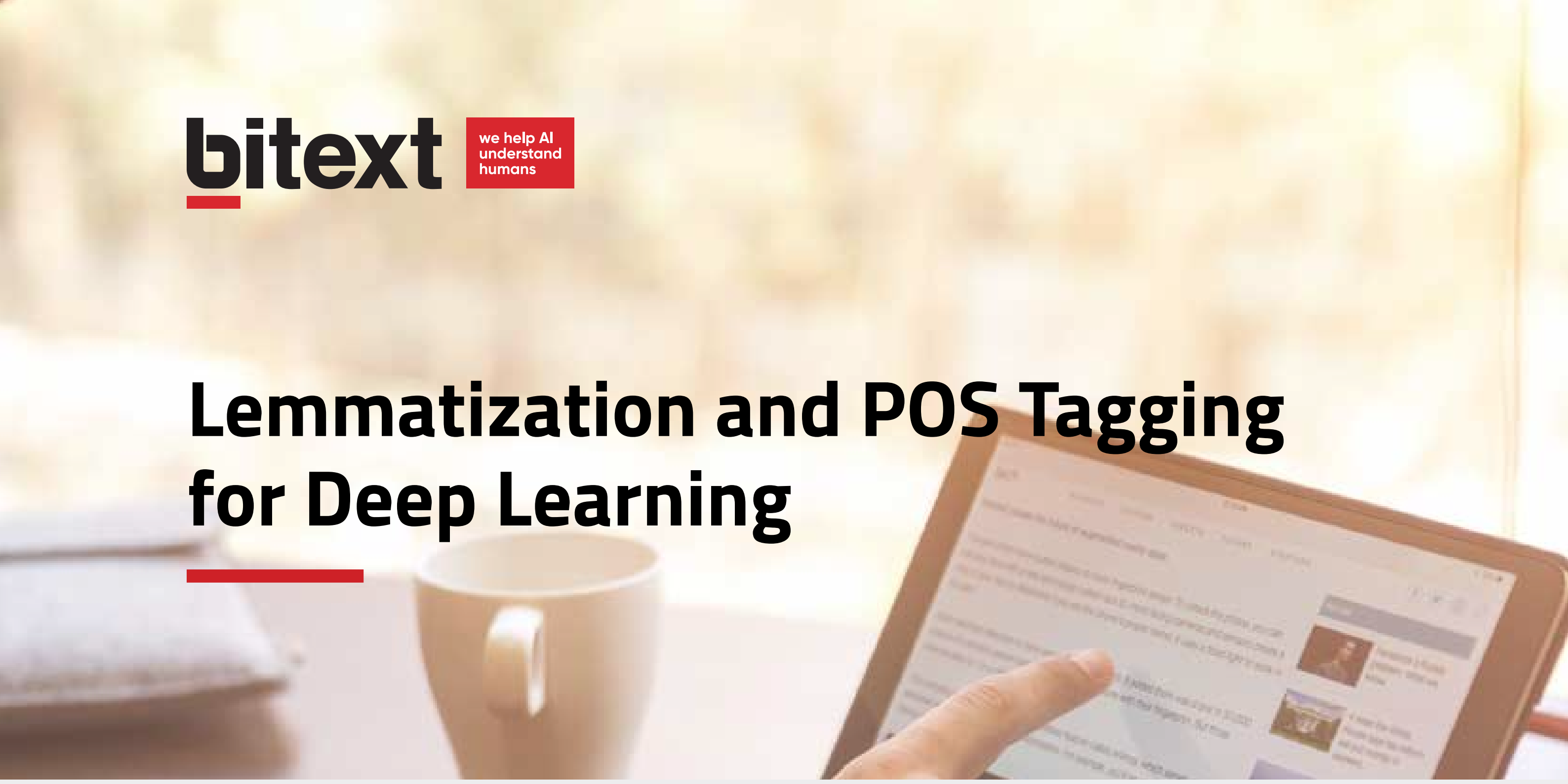

# Lemmatization and POS Tagging for Deep Learning

## 1 Introduction

*In the new era of Machine Learning, the Deep Learning era, the industry has proved able to solve some really challenging problems, even surpassing human performance sometimes. Modern Deep Learning algorithms, with enough data and computing power, can perform tasks impossible to be tackled only few years ago (e.g. in computing vision, banking or advertisement).*

*But nowadays only few problems allow end-to-end Deep Learning solutions where a single Neural Network is in charge of the entire job, using only raw data without preprocessing. Could hypothetically any problem be solved in this elegant way? In practice, for the majority of the problems, from autonomous cars to NLP problems, there is not enough data to try end-to-end solutions.*

*Specifically, for NLP problems, Deep Learning approaches typically use at least embeddings (word2vec or glove) to represent text: every word in the input text corresponds to a pretrained vectorized representation. So, in NLP engineers don't usually have an end-to-end solution, but at least a two-steps one, with embeddings preceding the main Neural Network.*

*The problem is that, in many fields, but specifically in those related to natural language, there is not enough data to learn from scratch that **superficially different inputs are equivalent** or that **superficially similar inputs are crucially different**. Hypothetically, with more data (and more computing power, if the job is to be done in time), Neural Networks would perform better and better, but the point is that engineers don't even have that data.*

*The idea of this paper is to explain how two preprocessing tasks (lemmatization and POS tagging) can be perform before the Deep Learning approach to still provide great results with less data and less time.*

# 2 Two main problems of natural languages

## Rich morphologies

As said above, superficially different inputs may be equivalent. An important example of this situation when dealing with natural languages are morphological changes over the same word. In natural languages, the same word can appear in texts with different forms. In English (a language with a poor morphology) the differences are not huge, but you can find interesting examples:

| Form | Example |
|---|---|
| **love** | *I **love** that camera that I used for years* |
| **loves** | *She **loves** that camera that I used for years* |
| **loved** | *She **loved** that camera that I used for years* |

In these examples, the single word “love” appears in three different forms: “love”, “loves” and “loved”. In other languages with a richer morphology (like Spanish or French) or even languages with a really complex one (like Finnish or Hungarian), there may be tens, hundreds or even thousands of different forms for the same word.

| English: 4 different verbal forms (example of the regular verb *love*) |
|---|
| love |
| loves |
| loved |
| loving |

| French: 34 different verbal forms (example of the regular verb *parler*) | | | |
|---|---|---|---|
| parle | parlaient | parlas | parlasiez |
| parles | parlerai | parla | parlassent |
| parlons | parleras | parlâmes | parlerais |
| parlez | parlera | parlâtes | parlerait |
| parlent | parlerons | parlèrent | parlerions |
| parlais | parlerez | parlasse | parleriez |
| parlait | parleront | parlasses | parleraient |
| parlions | parlé | parlât | |
| parliez | parlai | parlassions | |

| Hungarian: 95 different verbal forms (example of the regular verb *venni*, simplified here for practical reasons) | | | |
|---|---|---|---|
| veszek | veszegetted | vetetgetnéd | veszegettelek |
| veszel | vennél | vehessél | vennélek |
| vesz | vennéd | vehessed | veszegetnélek |
| veszünk | veszegetnél | vehetgessél | vegyelek |
| vesztek | veszegetnéd | vehetgessed | veszegesselek |
| vesznek | vegyél | vetessél | vehetlek |
| veszem | vegyed | vetessed | vehetgetlek |
| veszed | veszegessél | vetetgessél | vetetlek |
| veszi | veszegessed | vetetgessed | vetetgetlek |
| vesszük | vehetsz | vetethetsz | vehetnélek |
| veszitek | veheted | vetetheted | veszegethetnélek |
| veszik | veszegethetsz | vetegethetsz | vetetnélek |
| vettem | veszegetheted | vetetgetheted | vetetgetnélek |
| vettél | vetetsz | vehettetnél | vehesselek |
| vett | veteted | vehettetnéd | vehetgesselek |
| vettünk | vetegetsz | vehettetgetnél | vetesselek |
| vettétek | vetegeted | vehettetgetnéd | vetetgesselek |
| vették | vehetnél | vetethessél | vetethetlek |
| vetted | vehetnéd | vetethessed | vetetgethetlek |
| vette | veszegethetnél | vetetgethessél | vetethetnélek |
| vettük | veszegethetnéd | vetetgethessed | vetethetgetnélek |
| veszegetsz | vetetnél | veszlek | vetethesselek |
| veszegeted | vetetnéd | vettelek | vetetgethesselek |
| veszegettél | vetetgetnél | veszegetlek | |

This feature of natural languages makes the problem of data scarcity worse because **Deep Learning systems will find different textual units and learn their value, their impact in the solution of the task, separately**. It cannot benefit from the fact that all them are just the same word.

### Ambiguities

On the other hand, superficially similar inputs may be crucially different. This is what is called ambiguity. For example, when dealing with natural languages, a single form in a text could, potentially, correspond to different words. This is the case of the form "like" in English: it could correspond to either the verb "like" or the preposition "like":

| Form | Example |
|---|---|
| **Like** | *I **like** my new car because it's a hybrid* |
| **Like** | *My neighbor's car is **like** mine and sometimes I get confused* |

There are words in all languages that carry this ambiguity.

| Language | Form | POS | Meaning |
|---|---|---|---|
| Spanish | bajo | noun | hem / first floor |
| Spanish | bajo | adjective | short / low |
| Spanish | bajo | verb | I go down |
| Spanish | bajo | preposition | under |
| Greek | ότι | conjunction | that |
| Greek | ότι | determiner | any |
| Greek | ότι | pronoun | anything |
| German | Arm | noun | arm |
| German | Arm | adjective | poor |

This feature of natural languages also aggravates the problem of scarcity of data. **If different words with the same form have very different impact in the problem to be solved, it becomes necessary to disambiguate them well**. And, for that, the dataset would have to be big enough to contain a significant number of occurrences of the different values of that form.

## 3 Solution: lemmatization and POS tagging

Our solution for both challenges in projects related to natural languages is preprocessing the text in order to reduce its complexity and let the Deep Learning solution perform better and better just with the same amount of data. Besides, this approach will significantly reduce the number of epochs needed in the training process to converge, so hardware costs will also be reduced.

We provide two main techniques for that. They are compatible techniques and we strongly suggest combining them, in both the embeddings and the main Neural Network.

### Lemmatization

The first one is **lemmatization**, which can be used to normalize all different forms of the same word in the data according to its canonical version (its lemma). In the previous examples in English, all different occurrences of the word "love" ("love", "loves" and "loved") are reduced to the same canonical form "love".

| Form | Lemma | Lemmatized example |
|---|---|---|
| **love** | **love** | *I **love** that camera that I use for year* |
| **loves** | **love** | *She **love** that camera that I use for year* |
| **loved** | **love** | *She **love** that camera that I use for year* |

After this preprocessing step, the complexity of the data is reduced: **the system can now take all the occurrences of the same word (no matter its original form) and learn their impact (their value for the solution of the task) altogether.**

### Part of speech tagging (POS tagging)

The second technique we propose is **part of speech tagging**, which takes all the words in a text and tags them with its corresponding POS (its grammatical category, like "noun", "verb", "adjective", "preposition" and so on). This technique allows to disambiguate different words with the same form, such as "like", "bajo", "ότι" and "Arm" in the examples above.

| Form | POS | Example |
|---|---|---|
| **like** | **verb** | *I **like** my new car because it's a hybrid* |
| **like** | **preposition** | *My neighbor's car is **like** mine and sometimes I get confused* |

By applying this POS tagging process, the verb "like" has become distinguishable from the preposition "like". **The Deep Learning system doesn't need to learn those different values of the same form from context anymore**. And that distinction is crucial when those values have different impacts in the resolution of the problem.

## 4 Conclusion

Using Bitext services of lemmatization and POS tagging crucially reduces the amount of data and computing power needed to reach the desired results in any Deep Learning project related with natural language. Lemmatization lets learn the value of all occurrences of the same word altogether, and POS tagging disambiguates words with the same form but different impact in the resolution of the task.