fashion BRAIN project

Horizon 2020 Framework Programme

Grant Agreement: 732328 – FashionBrain

Document Information

Deliverable number: D2.1

Deliverable title: Named Entity Recognition and Linking methods

Deliverable description: As a result of task 1.1, this deliverable will consist of implemented algorithms for entity extraction from textual documents and linking to the ontology defined in WP1. The result will feed into WP 4, 5, and 6.

Due date of deliverable: 30.06.2018

Actual date of deliverable: 29.06.2018

Authors: Ines Arous, Mourad Khayati

Project partners: Beuth

Workpackage: WP2

Workpackage leader: UNIFR

Dissemination Level: Public

Change Log

Version	Date	Status	Author (Partner)	Description/Approval Level
1	24/06/18	Initial	Beuth	Public
2	29/06/18	Final	Beuth	Public

Contents

Abstract

In this deliverable, we implement a tool to identify fashion entities in a textual dataset and link them to a knowledge base such as Wikipedia. We describe the natural language processing (NLP) tools used to process a fashion dataset.

1 Introduction

Taxonomies give a clear and easy representation to understand a knowledge base. In the context of FashionBrain project, we have introduced a new taxonomy, called FashionBrain Taxonomy (FBT) and described in detail in D1.3. This taxonomy copes with the problem that the fashion taxonomies used by the fashion partners involved in the project are either inconsistent or have missing items. For example, items such "ear muffs", "skorts" or "flare jeans" do not exist in the Fashwell taxonomy but belong to other taxonomies. Thus, FBT provides a standard data representation schema of the knowledge bases used by the fashion partners. The proposed taxonomy merges Zalando's and Fashwell's taxonomies and extend them with items extracted from external sources such as Amazon or eBay. Even though FBT merges multiple taxonomies with new items constantly emerging, some fashion items are still missing in FBT. For example, none of the used taxonomies (including FBT) list "shawl" as a fashion item. These missing items need to be integrated to our pre-built taxonomy in order to enrich it.

The aim of this task is to identify these new items inside textual data sources and integrate them to the pre-built taxonomy. To solve this task, we proceed in a three-fold procedure:

1. Named Entity Recognition: Entity Recognition is the task of locating and classifying atomic elements in text into predefined categories [1]. In this step, we will locate key concepts inside textual data sources such as social network posts and classify them as fashion or non fashion concepts.

2. Named Entity Linking: Entity Linking is the task of assigning entities from a Knowledge Base to textual mentions of such entities in a document [2]. In this step, we will link the fashion concepts to their Wikipedia categorization.

3. Taxonomy Enrichment: In case the extracted concept does not exist in the taxonomy, we will add it to the most appropriate category in the FBT based on its Wikipedia categorization.

In this task, we have a extended an existing named entity recognition tool to fulfill the following desiderata:

- The tool should be able to recognize fashion items in any textual data source
- The tool should be able to maintain the FashionBrain taxonomy

The result of this task will help us enrich the pre-built taxonomy and will input to WP 4, 5 and 6

2 Background

In this section, we first describe the dataset used in the deliverable and then we describe the natural language processing (NLP) tools used to process a fashion dataset. More specifically, we describe in turn the Named Entity Recognition tool that we use to recognize entities in Fashion textual data and the Named Entity Linking technique used to match recognized entities to Wikipedia categorization.

2.1 Dataset

The used dataset contains information taken from 118 Instagram fashion influencer's accounts representing 2703 lines of text. For each fashion influencer, the pseudo name, the link to his profile picture, his biography and the most recent 100 posts are provided. Each post consists of a link to the published picture, the textual caption of the post together with some basic metrics as the number of comments and likes. We extract the textual data from this dataset which revolves around 100 textual captions of fashion influencer's posts and we apply the following NLP tools.

2.2 Named Entity Recognition (NER)

As a baseline for our work we use the state of the art NER framework *Stanford CoreNLP*[3]. Stanford CoreNLP is a Java annotation pipeline framework, which provides most of the common core natural language processing (NLP) steps [4]. We integrate the Stanford CoreNLP by including the **stanfordcorenlp** package in our tool. The CoreNLP toolkit provides several functionalities named "annotators". In our tool, we mainly use the following two annotators:

- The tokenization which is the task of chopping into pieces a textual document into tokens. A token is a sequence of characters which represents a semantic unit.
- Named Entity Recognition where the task is to extract so-called named entities from the text. Named entities are the chunks of words from text, which refer to an entity of certain type. CoreNLP is able to recognize named (PERSON, LOCATION, ORGANIZATION, MISC) and numerical (MONEY, NUMBER, DATE, TIME, DURATION, SET) entities. We use it mainly to check if a mentioned brand name is recognized as an organization.

Example 1 *Consider the example of this post: "Rewinding back to a few days ago when I went up into Coco Chanels apartment at Rue Cambon and had a lovely chat with @amandaharlech about #ChanelHandbagStories" The application of CoreNLP gives the results depicted in Figure 1:*

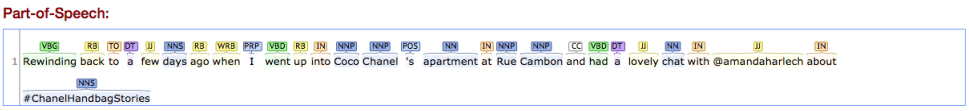


Figure 1: CoreNLP illustration.

Where "Coco Chanel" is recognized as an organization, "a few days ago" is recognized as Time and "Rue Cambon" is recognized as a location. However, CoreNLP is not able to identify "Handbag" as a concept. In this example, we are mainly interested in recognizing "Handbag" as entity.

In our tool, we tokenize our input data using Stanford CoreNLP and compare the results with those obtained with another Semantic/syntactic Extraction tool SENNA (cf. Section 3).

2.3 Named Entity Linking

An intuitive approach to perform named entity linking consists in applying a string matching approach. We use the Damerau-Levenstein distance to measure the distance between the recognized entity and Wikipedia concepts. The Levenstein distance between two strings is defined as the minimal number of edits required to convert one into the other. We use the Levenstein ratio which is derived from the Levenstein distance and varies between 0 and 1 where a value equals to 1 represents no required edits. The Levenstein ratio r between two strings a and b is defined as follows:

$$r = 1 - \frac{n}{(size(a) + size(b))}$$

We allow the so called "fuzzy" matching where we allow misspelled text to also be linked to the Wikipedia Knowledge Base.

Example 2 *Consider the fashion post introduced in Example 1. The Levenstein distance returns the number of edits n needed to convert a string a to string b. We need 13 edits to transform "ChanelHandbagStories" to "Handbag", the size of "ChanelHandbagStories" is 20 and the size of "Handbag" is 7. The Levenstein distance is the following:*

$$r = 1 - \frac{13}{7+20} = 0.52.$$

3 Tool

In this section, we describe our new natural language processing tool called FashionNLP which is specially designed for fashion textual data. This tool extends existing state of the art NER technique to fashion application. More specifically, FashionNLP

has three main components: NER, where fashion entities are recognized on a social media post, NEL, where we link the fashion entity to Wikipedia clothing ontology and finally, in case the fashion entity does not exist in the FashionBrain taxonomy, we add it to the taxonomy.

This section is organized as follows: We first explain in detail the FashionNLP pipeline. Then, we provide a quick start guide for the tool explaining how to install it and run it. Finally, we explain how we plan to extend our tool.

3.1 Pipeline

Figure 2 summarizes the pipeline of FashionNLP. We first start with the Named Entity Recognition (NER) step. This step consists in i) tokenizing the Instagram posts, tagging these posts, i.e., each word is identified as a noun, a verb, an adjective or a date and ii) identifying entities in predefined categories such as persons, organizations, locations, etc. We apply CoreNLP on Fashion influencers posts to perform the NER step.

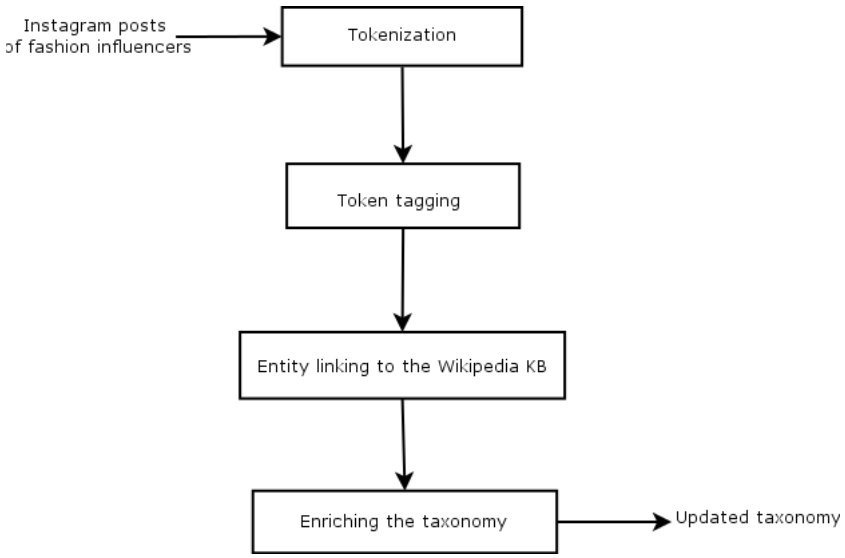


Figure 2: FashionNLP pipeline

We illustrate the NER step through an example. We process the following post "Shadow and Glow is a unique square shawl knitting pattern" using CoreNLP (and SENNA) as shown in table 3. The application of the two NER tools returns the same tagging and identification results but the term "Shadow" that is recognized by SENNA as a person while it is not recognized at all by CoreNLP. Both tools identify the entity "shawl" and tag it as a noun but none of them is able to recognize it as a fashion item (NER=O).

Thus, we propose to extend CoreNLP to be able to detect fashion items. We use *string matching* with the Levenstein distance and Wikipedia categorization as entity linking. More specifically, The Levenstein distance between "shawls" and "shawl" is -1/(length("shawl")+length("shawls")) and is equal to 0.909090909091. Using this distance, we are able to link "shawl", from the previous example, to the concept "Shawls and wraps"" (`https://en.wikipedia.org/wiki/Category:Shawls_and_wraps`) in the Wikipedia categorization.

Tokens	Tag	NER
Shadow	NNP	S-PER
and	CC	O
Glow	NNP	O
is	VBZ	O
a	DT	O
unique	JJ	O
square	JJ	O
shawl	**NN**	**O**
knitting	NN	O
pattern	NN	O
.	.	O

Table 1: SENNA

Tokens	Tag	NER
Shadow	NNP	O
and	CC	O
Glow	NNP	O
is	VBZ	O
a	DT	O
unique	JJ	O
square	NNP	O
shawl	**NN**	**O**
knitting	VBG	O
pattern	NN	O
.	.	O

Table 2: CoreNLP

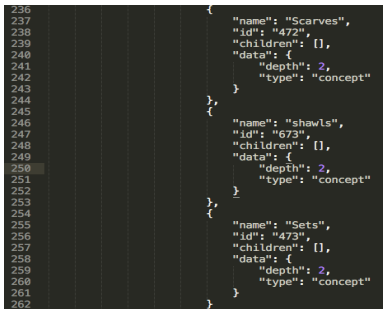
Table 3: Tokenization, token tagging and NER

```
['Shawls', 'Scarves'] ['scarves', 'scarves', 'scarves', 'scarves', 'sa
ves', 'shawl', 'shawl', 'shawl']
Item: shawls not found in FB taxonomy
```

Figure 3: Identification of “shawls” as a fashion entity

Finally, we enrich the FBT by adding the newly identified entity, “Shawls”, in case it does not already exist in the FBT taxonomy as shown in Figure 4.

(a) Taxonomy before the update.



(b) Updated Taxonomy.

Figure 4: Taxonomy Enrichment

3.2 Code

The source code is available through: `https://github.com/eXascaleInfolab/fashionNLP.git`

3.2.1 Installation

```
git clone https://github.com/eXascaleInfolab/fashionNLP.git
```

3.2.2 Description of the FashionNLP package

The **fashionnlp** package contains the following files:

- *updateFBT.py*: This code performs the following tasks:
  - Take a concept present in WikiKB as input.
  - Find the mentions of this concept and its similar concepts (using String matching and tree search) on instagram posts.
  - Find if the WikiKB concept is present in FBtaxonomy. If not, update the FB taxonomy.
- *wikitaxonomy.py*: This file is used to generate the wikipedia taxonomy from the wikipedia categorization `https://en.wikipedia.org/wiki/Category:Clothing_by_type`
- input folder: This folder contains the following input files:
  - *FBTaxonomy.csv*: The initial FashionBrain taxonomy in a csv format.
  - Find the mentions of this concept and its similar concepts (using String matching and tree search) in instagram posts.
  - *FBTaxonomy.csv*: The initial FashionBrain taxonomy in a json format.
  - *ner_posts.csv*: This file contains the output result of applying NER on the instagram posts.
  - *wikipediaKB.json*: This file contains the wikipedia knowledge base of fashion items in a json format.
- result folder: This folder contains the updated taxonomy

3.2.3 Running the code

Download StandfordCoreNLP from `https://stanfordnlp.github.io/CoreNLP/history.html` and add it to the cloning directory. In order to run an experiment, the updateFBT.py file is used. The corresponding command line to run the code is:

```
pip install stanfordcorenlp
pip install python-levenshtein
python updateFBT.py
```

4 Summary

In this deliverable, we have introduced a new natural language processing tool called FashionNLP, which is specially designed for fashion textual data. This tool extends existing state of the art NER technique, CoreNLP, to fashion application by applying string matching for entity linking. FashionNLP performs three steps introduced in the introduction namely i) named entity recognition, ii) named entity linking and iii) taxonomy enrichment. The results we have obtained are promising, but could be improved in multiple ways as described in the next section.

5 Next steps

The results we have obtained are promising, but could be further improved. For example, our technique is able to identify the category of jeans from the following post: "7/8 Jeans became very popular in 2018". However, it is not able to solve it yet because this category does not exist neither in the different fashion taxonomy we have access to nor in the Wikipedia taxonomy. This is due to the following reasons:

- Following the introduction of GDPR, the instagram API we are planning to use was blocked preventing us from getting access to fashion instagram posts. As a consequence, we have been using instagram posts that do not use enough references to fashion items to perform this task.
- Lack of annotated annotated fashion data to train the NLP tools.
- Our FashionBrain taxonomy does not list the fashion brand names inside a separate category.

In order to cope with the above-mentioned problems and improve the accuracy of the results, we propose to train a classifier to recognize fashion items on a social media post. For this task, first, we propose to manually annotate tokens on social media posts as fashion or non fashion tokens. Second, we aim to train a classifier using a recurrent neural network in order to identify fashion items in other datasets. In a later step, Word2Vec models trained on the generated fashion data can be used to identify emerging concepts and update the the FashionBrain taxonomy accordingly.

As this deliverable is related to Task 2.1 from Work Package 2 that runs until M24, we kindly ask for an extension of the deadline of this deliverable until M24 in order to be able to finish the tasks we are planning to perform in the context of this deliverable.

References

[1] S. Sekine and E. Ranchhod, *Named entities: recognition, classification and use*, vol. 19. John Benjamins Publishing, 2009.

[2] Z. Guo and D. Barbosa, "Robust entity linking via random walks," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pp. 499–508, ACM, 2014.

[3] "Stanford corenlp natural language software, 2018." Stanford CoreNLP, homepage: https://stanfordnlp.github.io/CoreNLP/.

[4] J. R. Finkel, T. Grenager, and C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling," in *Proceedings of the 43rd annual meeting on association for computational linguistics*, pp. 363–370, Association for Computational Linguistics, 2005.