Horizon 2020

Understanding Europe's Fashion Data Universe

Demo on Relation Extraction

Deliverable number: D4.4

Version 1.0

Funded by the European Union's Horizon 2020 research and innovation programme under Grant Agreement No. 732328

Project Acronym:	FashionBrain
Project Full Title:	Understanding Europe's Fashion Data Universe
Call:	H2020-ICT-2016-1
Topic:	ICT-14-2016-2017, Big Data PPP: Cross-sectorial and cross-lingual data integration and experimentation
Project URL:	https://fashionbrain-project.eu

Deliverable type	Report (R)
Dissemination level	Public (PU)
Contractual Delivery Date	31 December 2019
Actual Delivery Date	30 December 2019
Number of pages	16, the last one being no. 10
Authors	Alexander Löser, Tom Oberhauser, Benjamin Winter - BEUTH
Peer review	Alessandro Checco - USFD Alan Akbik - Zalando

Change Log

Version	Date	Status	Partner	Remarks
1.0	30/12/2019	Final	BEUTH,USFD	

Deliverable Description

This demo integrates methods for stacked deep learning on typical crowdbased workflows for trend detection and brand monitoring.

Abstract

Abstract

The goal of D4.4 is to demonstrate the relation extraction model reported on in deliverable D4.3. In D4.3 we analyzed three different possible approaches to relation extraction: Open Information Extraction (OpenIE) [4], SECTOR [1] and Hierarchical Reinforcement Learning for Relation Extraction (HRL-RE) [5]. From all evaluated methods, the Hierarchical Reinforcement Learning approach yields the best results leveraging strong transfer learning capabilities, Bidirectional Long Short-Term Memory (LSTM) and an elaborate learning scheme. Therefore, we demonstrate the HRL-RE algorithm in particular, since we deem it the most suited for this particular task and data.

We develop an intuitive Front-end Graphical User Interface (GUI) demonstrator for relation extraction and connect HRL-RE and the GUI using a common Application Programming Interface (API) based on the JSON variant of the TeXoo data model[1].

As discussed in D4.3 we do a further analysis on the relation extraction model using the newly crowd-sourced fashion dataset. This dataset was created in collaboration with University of Sheffield. It is based on the Fashion Communication Corpus of Hong Kong Polytechnic University (HKFCC) and the annotations stem from Amazon Mechanical Turk (www.mturk.com) (AMT).

[1]https://github.com/sebastianarnold/TeXoo

Table of Contents

List of Figures

List of Acronyms and Abbreviations

AMT	Amazon Mechanical Turk (www.mturk.com), micro-task crowdsourcing platform
API	Application Programming Interface
GUI	Graphical User Interface
LSTM	Long Short-Term Memory
NEL	Named Entity Linking
NER	Named Entity Recognition
REST	REpresentational State Transfer
HRL-RE	Hierarchical Reinforcement Learning for Relation Extraction
RE	Relation Extraction
OpenIE	Open Information Extraction
HKFCC	Fashion Communication Corpus of Hong Kong Polytechnic University
QA	Question Answering

1 Introduction

The FashionBrain project targets at consolidating and extending existing European technologies in the area of database management, data mining, machine learning, image processing, information retrieval, and crowd-sourcing to strengthen the positions of European (fashion) retailers among their world-wide competitors.

Already, D1.2 unraveled in interviews with experts, that one important goal of fashion retailers is to be able to extract, analyze, and predict fashion trends from largely unstructured data. One way to do that is to extract fashion specific entities and relations from text sources like news articles, blog posts and social media streams.

To that end, in D4.3 "Relation Extraction with Stacked Deep Learning" we developed and applied deep learning models to the task of relation extraction and evaluate them with regard to their usefulness in the fashion domain. Since relation extraction is a particularly hard and still unsolved task, we investigated three approaches:

(1) Open Information Extraction (OpenIE) applies generic lexico-syntactic patterns to detect un-typed relation candidates. We benchmark four leading systems.

(2) We re-implement and benchmark a state-of-the-art supervised-learning approach from [5]. The approach is based on hierarchical reinforcement learning and Bidirectional-LSTMs. This model in particular, HRL-RE, solves not only the task of binary relation extraction, but also the task of Named Entity Recognition, which is a prerequisite task for relation extraction, in a single system.

(3) We design and prototype an approach that can assign relation types across sentences and even for entire paragraphs. The approach, called SECTOR, can be trained semi-supervised and alone from information in headlines and text related to headlines. Thereby this approach, like approach (1) circumvents the requirements of (2) of having large sets of training data, which are not available in the fashion domain.

This report demonstrates one of these approaches to relation extraction, HRL-RE, using data from the fashion domain. In Section 2.1 we illustrate the Front-end user interface of this demonstrator, Section 2.2 depicts the API that is used for communication between the Front-end and our model, Section 2.3 describes the data collection and aggregation process and finally Section 2.4 shows the new results on the crowd-sourced fashion data.

1.1 Scope of This Deliverable

As Core Technology CT4, this deliverable fits into the execution layer of this projects' deliverables. We add our models to the library of trained models of this project, therefore extending Deliverable D2.5 "Library of trained Deep Learning models". The prerequisite Named Entity Recognition (NER) task, which is currently handled by HRL-RE itself, can be substituted by other NER models already developed in the scope of this project. In particular, D2.1 "Named Entity Recognition and Linking method" resulted in models and algorithms that this deliverable can build on top off. We further collaborate with the University of Sheffield in the scope of deliverable D3.3 "Surveys design and crowd-sourcing tasks" to crowd-source a novel fashion themed relation extraction dataset that we discussed in more detail in D4.3. A more detailed analysis of the biases resulting from the aggregation methods is presented in D3.4. Lastly, it should be evaluated whether this deliverable is already fitting for deliverable D5.3 "The classification algorithm and its evaluation on fashion time series" and D5.4 "Demo on Fashion Trend Prediction", as we can also extract dates and time series, possibly even timestamps.

The results of our work in T4.3 are reported in two deliverables. The last deliverable, D4.3 "Relation Extraction with Stacked Deep Learning", contained the theoretical description of the method which is implemented and demonstrated in this deliverable D4.4. Both deliverables also contain experiments showing this architectures' effectiveness, D4.3 using standard benchmark corpora, and D4.4 using actual fashion data.

2 Demo on Relation Extraction

This chapter provides a detailed summary of the application of our trained relation extraction model on standard benchmark corpora, as well as actual fashion data. Section 2.1 introduces the GUI, which allows an interactive exploration of the models' annotation capabilities. Further, Section 2.2 describes the API that the front-end uses for communication with our model back-end. Finally, Section 2.3 introduces our crowd-sourced fashion corpus and Section 2.4 closes with an evaluation of the relation extraction model on it.

2.1 Graphical User Interface

The graphical user interface is a web-based front-end. It is based on the frameworks Vue.js[1], Vuetify[2], Webpack[3] and Axios[4]. Furthermore, we built it as a standalone component that we not only develop for use as a GUI for relations extraction, but also NER, Named Entity Linking (NEL) and Question Answering (QA). Moreover, this GUI can eventually also be used as an annotation tool to collect further data in the future, comparable to the one that was used by the University of Sheffield in connection with AMT. The start-page of the prediction interface is shown in Figure 2.1. It enables a user to past a sample of text and to apply the Relation Extractor by pressing the "Annotate" button. The text sample is then sent to the API, which we describe in Section 2.2. After the API finished the annotation job, it's results will be displayed as cards as depicted in Figure 2.2 which show the predicted relation type, as well as the extracted entities.

[1]https://vuejs.org/
[2]https://vuetifyjs.com/
[3]https://webpack.js.org/
[4]https://github.com/axios/axios

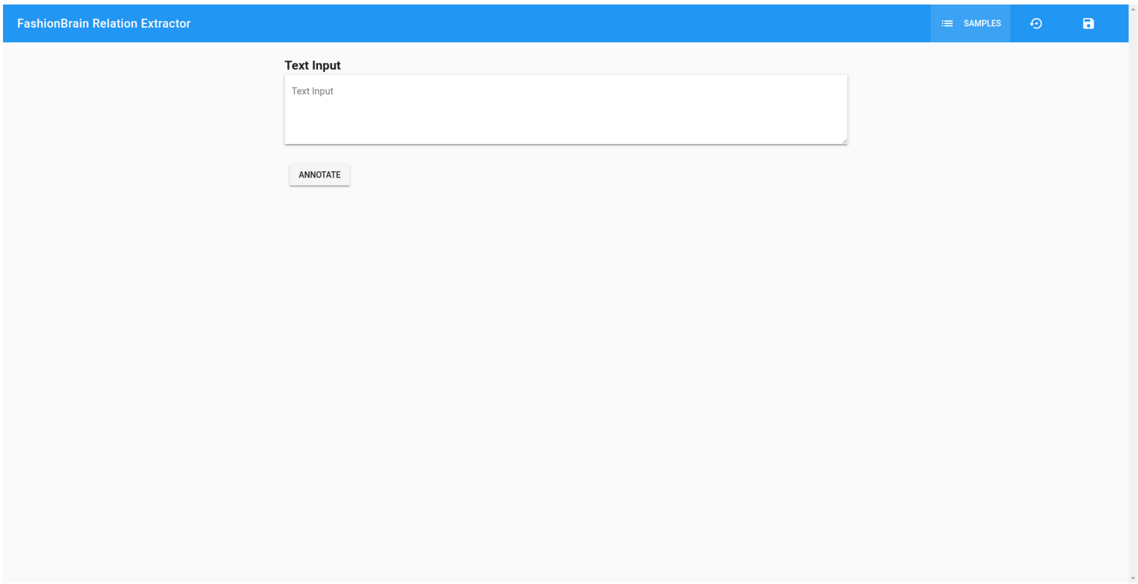

Figure 2.1: The prediction interface of the FashionBrain Relation Extractor, consisting mainly out of a text input field and the annotate button.

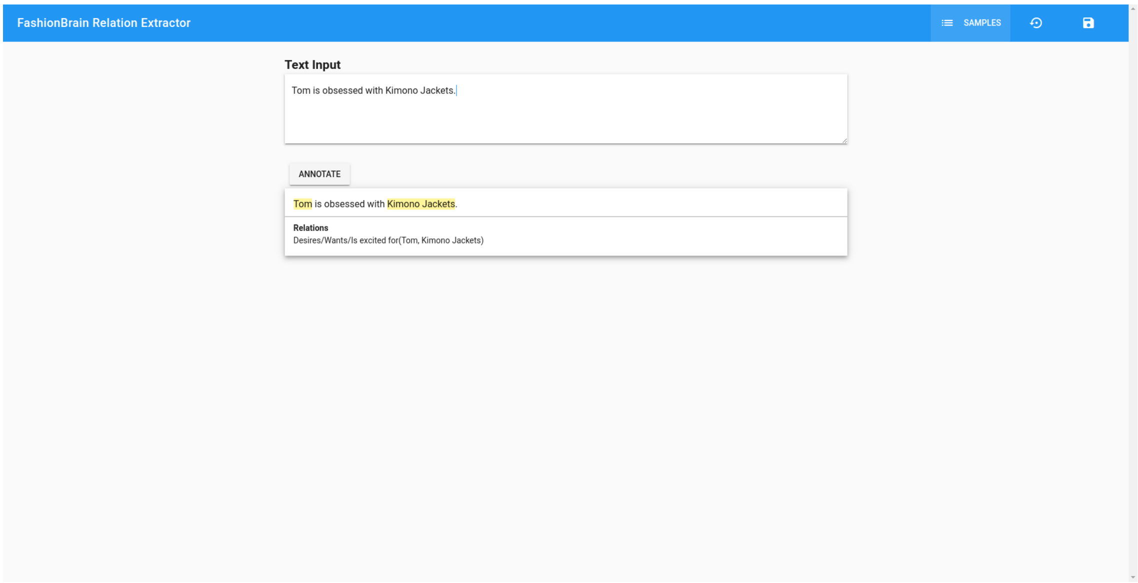

Figure 2.2: The prediction interface, after a text has been inserted and a prediction has been made. A card shows both the predicted relation and the extracted entities.

2.2 API

Our API consists of a single REpresentational State Transfer (REST) connection endpoint. While the front-end uses Axios to send a REST based request to our back-end, we use the Python module Flask to set up a minimal server as our back-end. This server, for now, only has a singular route expecting GET requests containing text to be annotated in the particular JSON format depicted in Figure 2.3. The server then splits the text by sentences and proceeds with the necessary pre-processing. Our relation extraction model then annotates all of the sentences and returns a JSON object in the TeXoo data format. An example is depicted in Figure 2.4.

```
{"text": "Tom is obsessed with Kimono Jackets."}
```

Figure 2.3: Example of a text in the JSON format

```
{
  "length": 36, "begin": 0, "class": "Sample", "type": null, "tokens": null,
  "empty": null, "language": null, "sentences": null, "source": null,
  "id": null, "title": "", "documentRef": null, "uid": null,
  "text": 'Tom is obsessed with Kimono Jackets.',
  "annotations": [{
    "length": null, "uid": null, "text": null, "begin": null, "type": null,
    "source": null, "confidence": null, "isActive": false, "documentRef": null,
    "classs": "relationAnnotation",
    "predicate": "desires/wants/is_excited",
    "relationArguments": [{
        "length": 3,
        "documentRef": null,
        "uid": null,
        "text": "Tom",
        "begin": 0,
        "classs": "relationArgument",
        "type": "GENERIC",
        "source": "GOLD",
        "confidence": null,
        "isActive": false
      },
      {
        "length": 14,
        "documentRef": null,
        "uid": null,
        "text": "Kimono Jackets",
        "begin": 21,
        "classs": "relationArgument",
        "type": "GENERIC",
        "source": "GOLD",
        "confidence": null,
        "isActive": false
      }
    ]
  }]
}
```


Figure 2.4: Example of the annotated text output JSON format

2.3 Data Collection and Aggregation

To annotate the Fashion Corpus, we had to consider the complexity of the annotation process. For this reason we first prepared a qualification task to select suitable Amazon Mechanical Turk (AMT) workers. Then we performed a two-phase data annotation as follows:

Classification: each text fragment have been classified by multiple workers. The results have then been aggregated as described in Section 2.3.1.

Text highlighting: using the results of the aggregation, we ask workers to highlight the text fragments corresponding to the different parts of the relation. This part is explained in detail in Section 3.2 of Deliverable D4.3.

The novelty of this method is on the aggregation method, that we explain in detail in the next section.

2.3.1 Data Aggregation - Simplified Dawid Skene Model

Using crowdworkers for data aggregation requires to perform an estimation of the biases and worker abilities [3]. While a Bayesian equivalent of the Dawid-Skene model [2] is usually a good choice to perform annotation aggregation, in crowdsourcing often many different workers perform the annotation, making the estimation of the real confusion matrix (per-worker and per-category bias) unfeasible, as it would require a number of parameters comparable to the number of observations.

For this reasons, we reduced the degrees of freedom of the model by considering a one parameter only confusion matrix $\alpha_i \cdot M$, where M is a $c \times c$ identity matrix (and c is the number of categories), and α is the ability of the worker i. We refer to Deliverable D3.4 for more details on this model.

2.4 Evaluation on Crowd-Sourced Fashion Corpus

We evaluate the HRL algorithm on the crowd-sourced fashion corpus in multiple settings, in particular regarding the transfer learning mode. First, we evaluate HRL without using any transfer learning at all, giving us a baseline on how difficult this corpus is to solve. After that, we evaluate three different transfer learning scenarios, all of which are based on pre-training on the NYT11 benchmark corpus used in Deliverable D4.3. The four approaches are therefore as follows:

- Baseline: Do supervised pre-training on HKFCC, then so reinforcement learning training on HKFCC
- TL1: Do supervised pre-training on NYT11, and finally do reinforcement learning training on HKFCC
- TL2: Do supervised pre-training on NYT11, then do supervised pre-training on HKFCC, and finally do reinforcement learning training on HKFCC
- TL3: Do supervised pre-training on NYT11, then do reinforcement learning training on NYT11, and finally do reinforcement learning training on HKFCC

We measure the combined F1 value for both, the NER and the Relation Extraction (RE) stage, as using both in unison is the main mode of the algorithm and they are heavily dependent on each other.

The results show that transfer learning can indeed bring moderate improvements over just using the HKFCC corpus by itself. This is to be expected, since the

Approach	Dev-Set F1-Score
Baseline	0.18
TL1	0.19
TL2	0.25
TL3	0.09

crowd-sourced dataset is very small, so using additional data is very helpful The biggest improvement brings TL2, the supervised pre-training on NYT11 seems to transfer very well to supervised pre-training on HKFCC. TL1 only brings a very small improvement since it seems, that HKFCC and NYT11 seem to be too different from each other for simple pre-training on NYT11 to suffice for a transfer to the downstream task reinforcement learning training. Somewhat surprisingly TL3 even has a decidedly negative effect on the downstream performance. One reason for that could be that the reinforcement learning training overfits too much on the specific task, getting into deep local minima, making it generalize less optimally.

3 Conclusion

This demo shows the application of a state-of-the-art relation extraction approach on the task of fashion data relation extraction. The relation extractor makes use of stacked deep learning and leverages transfer learning to make the most out of the small amount of available training data. Further, we compensated the lack of in-domain training data by utilizing crowd-sourcing technologies. We built an intuitive web-based front-end to allow users interactively explore the models' annotation capabilities. This front-end communicates with our model through JSON objects in the TeXoo data format. This deliverable also evaluates our model on the newly crowd-sourced fashion dataset and shows that the combination of pre-training on NYT11, pre-training on HKFCC with a final reinforcement learning step on HKFCC is the best method of transfer learning for this task. However, the results are still worse than those of the benchmark corpus presented in Deliverable D4.3, due to a combination of the difficult and particular fashion language and the only very small size of the dataset.

Bibliography

[1] Sebastian Arnold, Rudolf Schneider, Felix A Gers, Philippe Cudré-Mauroux, and Alexander Löser. SECTOR: A Neural Model for Joint Segmentation and Topic Classification. *TACL*, (to appear):12, 2019. 00000.

[2] Alexander Philip Dawid and Allan M Skene. Maximum likelihood estimation of observer error-rates using the EM algorithm. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 28(1):20–28, 1979.

[3] Panagiotis G Ipeirotis, Foster Provost, and Jing Wang. Quality management on amazon mechanical turk. In *Proceedings of the ACM SIGKDD workshop on human computation*, pages 64–67. ACM, 2010.

[4] Rudolf Schneider, Tom Oberhauser, Tobias Klatt, Felix A. Gers, and Alexander Löser. Analysing errors of open information extraction systems. In *Building Linguistically Generalizable NLP Systems*, Copenhagen, Denmark, 2017. 00005.

[5] Ryuichi Takanobu, Tianyang Zhang, Jiexi Liu, and Minlie Huang. A hierarchical framework for relation extraction with reinforcement learning. *CoRR*, abs/1811.03925, 2018. URL http://arxiv.org/abs/1811.03925.