Horizon 2020

Understanding Europe's Fashion Data Universe

Relation Extraction with Stacked Deep Learning

Deliverable number: D4.3

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	30 June 2019
Actual Delivery Date	29 June 2019
Number of pages	32, the last one being no. 26
Authors	Alexander Löser, Tom Oberhauser, Benjamin Winter - BEUTH
Peer review	Mourad Khayati - UNIFR Alessandro Checco, Kathryn Mackellar - USFD

Change Log

Version	Date	Status	Partner	Remarks
1.0	29/06/2019	Final	BEUTH,USFD	

Deliverable Description

This report integrates relation extraction and stacked deep learning for fashion related relations. We will investigate different approaches and how much of the training should be executed in the database or how much may be shipped to a less expensive GPU-based architecture.

Abstract

The goals of this deliverable are to examine approaches for relation extraction from text data. In this report we present three different approaches to this task: Open Information Extraction (OpenIE), SECTOR, and Hierarchical Reinforcement Learning for Relation Extraction (HRL-RE) which we implement within the FashionBrain project and which we evaluate regarding the fashion domain.

The task of relation extraction is a particularly hard and still mostly unsolved problem and we are not aware of any 'golden bullet' that will fit the project. Therefore, each of the three proposed approaches has strengths and weaknesses.

1. OpenIE is an entirely unsupervised method of extracting relation predicates. Being unsupervised is very useful in a domain where few to no data exists, such as the fashion domain. We compared four state-of-the-art approaches of major systems [11].
2. We tested a state-of-the-art system for binary relation extraction [14] from supervised data. The Hierarchical Reinforcement Learning approach yields the best results leveraging strong transfer learning capabilities, Bidirectional Long Short-Term Memory (LSTM) and a complex learning scheme.
3. We introduce a novel approach, SECTOR (to appear at TACL 2019) [3], which benefits in a self-supervised fashion from existing data e.g. Wikipedia. SECTOR predicts which topic (relation) a sentence or paragraph belongs to. Thereby, SECTOR relaxes the assumption that a relation must be extracted based on spans of arguments in the same sentence and with that removes one of the dominant error classes in relation extraction.

Because no fashion training data exists, we pursue two different approaches to generating such data. Firstly, we generate a new non-fashion benchmark corpus, that is a combination of standard corpora. In this corpus we select relations that most closely resemble the fashion domain to create a dataset that is amenable to transfer learning. Through transfer learning our model is able to learn more efficiently from smaller amounts of data generated by our second approach. The second approach is to collaborate with University of Sheffield on crowd sourcing a fashion themed relation extraction dataset. This dataset, to the best of our knowledge, is the first of its kind.

In our future work in D4.4 we will propose a intuitive user interface that illustrates relations spotted in text.

Table of Contents

List of Figures v

List of Tables v

List of Acronyms and Abbreviations vi

1 Introduction 1
- 1.1 Definition of Relation Extraction . . . 2
- 1.2 Scope . . . 2
  - 1.2.1 Scope and Dependencies . . . 2
  - 1.2.2 Analysis of Six Existing State-of-the-Art Datasets . . . 3

2 Three Approaches to Relation Extraction 5
- 2.1 Open Information Extraction . . . 5
  - 2.1.1 Quantitative Evaluation of OpenIE Systems . . . 6
  - 2.1.2 Qualitative Assessment of Common Error Classes . . . 7
  - 2.1.3 Recommendation . . . 8
- 2.2 SECTOR . . . 8
  - 2.2.1 Paragraph Classification . . . 9
  - 2.2.2 Recommendation . . . 10
- 2.3 HRL-RE . . . 11
  - 2.3.1 High-Level Policy . . . 11
  - 2.3.2 Low-Level Policy . . . 12
  - 2.3.3 Model and Training Scheme . . . 13
  - 2.3.4 API . . . 13
  - 2.3.5 Recommendation . . . 14

3 Discussion 18
- 3.1 Improving the Model of HRL-RE . . . 18
- 3.2 Curating Own Fashion Training Data . . . 19
  - 3.2.1 Crowdsourced Annotations for the Fashion Communication Corpus . . . 20

4 Conclusions 24

Bibliography 25

List of Figures

2.1 Example of an OpenIE extraction 5
2.2 Results of the quantitative evaluation of four OpenIE systems 6
2.3 Results of the qualitative evaluation of four OpenIE systems 7
2.4 WikiSection task 9
2.5 Overview of the hierarchical relation extraction process 11
2.6 Example of a relation in the TeXoo JSON format. 14

3.1 HKFCC annotation task (1) 21
3.2 HKFCC annotation task (2) 22
3.3 HKFCC annotation task (3) 22
3.4 HKFCC annotation task (4) 23

List of Tables

2.1 Qualitative error categories for OpenIE systems 7
2.2 Relation Classes in Combination Corpus 15

List of Acronyms and Abbreviations

API	Application Programming Interface
JSON	JavaScript Object Notation
LSTM	Long Short-Term Memory
NER	Named Entity Recognition
NLP	Natural Language Processing
REST	REpresentational State Transfer
HRL-RE	Hierarchical Reinforcement Learning for Relation Extraction
RE	Relation Extraction
OpenIE	Open Information Extraction
POS	Part-of-Speech
HKFCC	Fashion Communication Corpus of Hong Kong Polytechnic University
MDP	Markov Decision Process

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 crowdsourcing 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 develop and apply 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 investigate 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 [14]. 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 discusses our approaches to solve relation extraction in the fashion domain in Section 2 and specifically Section 2.1, Section 2.2 and Section 2.3. We then do an analysis of our last approach in Section 2.3.5 and outline possible future work and extensions in Section 3.

1.1 Definition of Relation Extraction

Relation Extraction (RE) is the task of extracting and classifying relations between entities in unstructured data like texts. Gaining information on the relations between entities is a crucial step for many downstream tasks like knowledge-base population or question answering. Most RE systems extract relations that are strictly typed and consist of a type and two (binary) arguments. More formally, such systems yield tuples $r(e_1, e_2)$ where $r \in R$ is the relation type out of a set of relation types R and $e_{1,2}$ are the entities that are connected by the relation. Such binary relations however tend to lose their informativeness due to context loss, so advanced systems also need the capability of extracting N-ary relations [1]. Another problem a RE system faces is the problem of co-references, especially those that span over multiple sentences.

Besides the typed Relation Extraction, there are also untyped approaches like OpenIE. The Open IE paradigm was introduced by Banko et al. in 2007 and describes an unsupervised extraction mechanism solely based on the linguistic features of a text [4]. A detailed analysis of the applicability of Open IE systems to our task is provided in section 2.1. Despite decades of research, the task of relation extraction remains a difficult problem. Due to the complexity of the matter, relation extraction systems may achieve satisfying results for certain use-cases but require high-quality and domain specific training sets that are difficult to obtain.

1.2 Scope

In the context of T4.3 “Interactive stacked deep learning for crowdsourcing”, we discuss three approaches, OpenIE, SECTOR, and HRL-RE. The task of relation extraction and these three approaches specifically target the following business scenarios defined by deliverable D1.2.

- Scenario 3 “Brand Monitoring for Internal Stakeholders ”and Challenge 5 “Opinion Mining on Fashion Reviews” in particular. Extracting useful representations of customer opinion is a difficult task. It is helped by first extracting the entities and their relations the customer is talking about.
- Scenario 4 “Fashion Trends Analysis” As described in the introduction, extracting relationships between e.g. products, companies, trend setters is an important step to monitoring trends in the fashion domain.

1.2.1 Scope and Dependencies

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 University of Sheffield in the scope of deliverable D3.3 "Surveys design and crowdsourcing tasks" to crowdsource a novel fashion themed relation extraction dataset that is discussed in more detail in the following section. 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. This deliverable, D4.3 "Relation Extraction with Stacked Deep Learning", contains the theoretical description of the method which is implemented and demonstrated in deliverable D4.4 (due in project month 36) and also experiments showing this architectures' effectiveness.

1.2.2 Analysis of Six Existing State-of-the-Art Datasets

This deliverable and its task of relation extraction require a very specific type of dataset. Such a dataset would need to map both named entities, and the relations between them, to text sequences. Unfortunately, to the best of our knowledge, no such dataset is currently available. A simple ontology or taxonomy as developed in deliverable D1.3 "Develop ontology/taxonomy for partners to share data" is unfortunately not sufficient for this task, as it maps only entities not the relations between them. Like the previous deliverables in this work package, this deliverable does not originally profit from datasets created by D3.3 either.

As a fashion themed relation extraction dataset does not currently exist, this deliverable, and deliverable D4.4 by extension, focus on implementing and validating a system that can be compared to the state-of-the-art, and is applicable to standard benchmark datasets that match the fashion domain as closely as possible. We further leverage multiple standard corpora to create a new corpus, which models the fashion domain better than any single one of them. In order to create this dataset, we evaluate the following datasets: TACRED, NYT10, SEMEVAL 2007, SEMEVAL 2010, FewRel and ACE05. Unfortunately SEMEVAL 2007, FewRel and ACE05 are not fit for the purpose of our dataset as they all share one or both of the following properties:

- There exist too few examples (tens instead of many hundreds or even thousands) per each class / relation type. Since we aim to heavily rely on the transfer learning capabilities of deep models, classes with only few samples are not very useful.

- Classes / relation types and entities are too general and don't map well onto the fashion domain. Again, due to our transfer learning approach it is necessary to learn from classes which closely match the target domain.

That leaves the datasets TACRED, NYT10 and SEMEVAL 2010. We manually filter and aggregate them by relation type and leave in mainly those relations, that map well to the fashion domain. We add some larger non fashion classes, in order to improve generalization. The resulting dataset contains relations pertaining to two main classes of entities: Companies, which would be Fashion Retailers like Zalando, and Persons, which relates to celebrities, CEOs or other trendsetters. Some examples include the relation types "Product-Producer", "Organization-Subsidiaries" and "Person-EmployeeOf". Overall, in Table 2.2 we examined 81 relation types in precison, recall and F1. Please refer to Section 2.3.5 for further details.

2 Three Approaches to Relation Extraction

This section provides a detailed summary of three different approaches that we evaluated with focus on their applicability to the task of relation extraction in the fashion domain. As stated in section 1.2, there is few training data available in the fashion domain, especially for the supervised training of RE systems. Therefore we start by evaluating the unsupervised approach of OpenIE at first. Following the lessons learned, we continue with the evaluation of two other systems that may either be used for (SECTOR [3]), or are especially designed to yield typed relation tuples (HRL-RE [14]).

2.1 Open Information Extraction

The goal of OpenIE systems is to extract relation tuples in an unsupervised manner. In contrast to traditional RE systems, their output is not typed e.g. $r_t(e_1, e_2)$ but consists of arguments and their connecting predicate $(e_1, predicate, e_2)$. Figure 2.1 shows an example of an OpenIE relation with one predicate and three arguments. Depending on the type of downstream task, the fact that those systems are not restricted to a fixed schema can be a core advantage or a major downside. However, the main advantage of such systems, especially in domains where training data is sparse, is the unsupervised extraction of relations. OpenIE systems extract relations by classifying the parts of it by applying rule based or machine learning methods on the linguistic structures of the text like dependency parses or shallow Part-of-Speech (POS) tags.

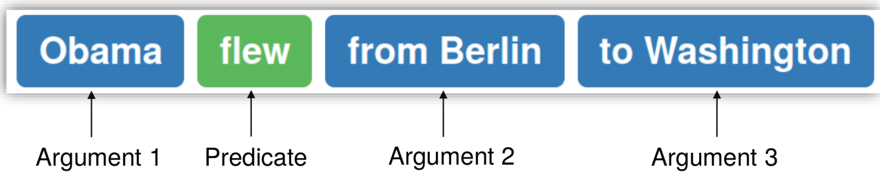

Figure 2.1: Example of an n-ary OpenIE extraction with one predicate and three arguments.

2.1.0.1 Stanford OpenIE, OpenIE 4.7, CLAUSIE and PredPatt

We conducted an in-depth analysis of 4 different OpenIE systems, namely *Stanford OpenIE (SIE)* [2], *OpenIE 4.2 (OIE)*[1], *ClausIE (CIE)* [5] and *PredPatt (PP)* [15]. The performance of each OpenIE system was quantitatively evaluated on four carefully picked relation rxtraction tasks, namely *NYT-222*, *WEB-500* [7], *PENN-100* [16] and *OIE2016* [12]. In addition, all systems were qualitatively evaluated with respect to a list of common errorclasses that was compiled from recent literature [11].

2.1.1 Quantitative Evaluation of OpenIE Systems

The quantitative measurements *Precision*, *Recall* and F_2 for each system, dataset and text-matching strategy are shown in Table 2.2. In contrast to most literature, the table displayed in figure 2.2 reports the F_2 instead of the F_1 score because F_2 gives more weight to the recall. Because of OpenIE systems being commonly used as intermediate step towards another downstream task, a higher recall is more important than a higher precision because the downstream application may filter further and would benefit from more data. We compared two different text-matching strategies (strict containment (a) vs. relaxed containment (b)) that lead to more or less strict interpretation of a correct extraction result.

Dataset	ClausIE (%)			OpenIE 4.2 (%)			Stanford OIE (%)			PredPatt (%)		
	P	R	F_2	P	R	F_2	P	R	F_2	P	R	F_2
PENN-100 (a)	4.00	21.15	11.39	12.41	36.54	26.31	**14.85**	**57.69**	**36.58**	6.83	42.30	20.75
PENN-100 (b)	4.00	21.15	11.39	13.07	38.46	27.70	**14.85**	**57.69**	**36.59**	7.76	48.08	23.58
WEB-500 (a)	**16.33**	**46.70**	**34.03**	12.83	19.62	17.74	13.65	40.72	29.16	5.18	13.43	10.19
WEB-500 (b)	**16.33**	**46.70**	**34.03**	13.39	20.47	18.51	13.65	40.72	29.16	6.09	15.78	11.97
NYT-222 (a)	1.64	5.85	3.87	**2.86**	7.66	5.73	0	0	0	2.22	**13.51**	**6.71**
NYT-222 (b)	4.69	16.67	11.03	11.28	30.18	22.60	**13.37**	**73.87**	**38.77**	8.47	51.35	25.51
OIE2016 (a)	14.81	13.67	13.89	**24.85**	**18.69**	**19.67**	0.80	1.49	1.27	7.26	12.39	10.86
OIE2016 (b)	20.38	18.81	19.10	**39.58**	**29.76**	**31.31**	3.83	7.10	6.07	13.52	23.09	20.23

Figure 2.2: Quantitative results of the evaluation of the four OpenIE systems. (a) and (b) denote different matching standards where (b) is more relaxed than (a) [11].

Table 2.2 clearly shows that OpenIE systems, when measured against classic relation extraction tasks, do not produce impressive results. Besides recall, where especially *Stanford OpenIE* produces at least reasonable results on the *PENN-100* and *NYT-222* dataset, all systems suffer from low precision values.

[1] https://github.com/knowitall/openie

2.1.2 Qualitative Assessment of Common Error Classes

In addition to the comprehensive quantitative analysis, we also performed a qualitative evaluation of the systems. Each system was presented a subset of 17 sentences from each dataset. The predictions performed by each system were qualitatively categorized into error categories (see table 2.1) by two independent judges.

Table 2.1: Qualitative error categories for OpenIE systems

Wrong Boundaries	Boundaries of argument or predicate are too large or too small.
Redundant Extraction	Multiple extractions for the same sentence and subject-predicate structure.
Uninformative Extraction	Following Fader et al. [6], uninformative extractions are extractions that omit critical information.
Missing Extraction	A relation that was present in the dataset but not found by the OpenIE system.
Wrong Extraction	A strict containment strategy does not yield a match for the predicate and all arguments and / or the number of arguments does not match.
Out of Scope	The OpenIE system predicted a relation that was marked by both judges as a valid relation with information gain, but it was not present in the dataset.

Dataset	**NYT-222** (n-ary)				**OIE2016** (n-ary)				**PENN-100** (binary)				**WEB-500** (binary)			
# Relations	17				29				17				17			
	CIE	OIE	PP	SIE	CIE	OIE	PP	SIE	CIE	OIE	PP	SIE	CIE	OIE	PP	SIE
# Predicted	42	35	68	74	28	30	57	91	63	34	61	49	33	22	24	38
# Correct	2	1	**6**	0	8	**12**	6	5	4	8	10	**11**	5	4	3	**10**
# Redundant	0	0	0	**5**	0	0	0	**18**	1	0	0	**4**	**2**	0	0	0
# Uninformative	4	2	**8**	0	2	0	**6**	1	**9**	3	**9**	4	0	0	0	**3**
# Boundaries	11	17	18	**39**	11	11	21	**69**	**14**	5	9	**14**	8	**9**	**9**	**9**
# Wrong	2	1	3	**5**	1	1	**6**	3	3	1	**10**	4	1	**2**	**2**	**2**
# Out of Scope	24	17	**34**	30	7	6	**21**	13	**33**	17	31	18	**19**	8	12	14
# Missed	4	1	**5**	**5**	8	4	7	**12**	**14**	6	6	7	8	3	**11**	6

Figure 2.3: Results of the qualitative evaluation of the four OpenIE systems [11] From each dataset, 17 sentences were used and a total of 749 predictions were evaluated in total. All datasets, except *OIE2016*, contained one relation per sentence to be found.

Figure 2.3 provides an overview of the evaluation results. In total, 749 predicted extractions were evaluated. When looking at the first content row, it is noticeable that, even though the evaluation data contained 17 or 29 relations to find, most systems predicted way more relations than required. This behaviour is best explained by the number of "Out of Scope" errors. Often, the dataset itself provides only one relation per sentence but the sentence contained more extractable relations.

Besides "Out of Scope" errors, the most salient category is the "Wrong Boundaries" error group among all datasets. We observed that all OpenIE systems often produce boundary errors, mostly due to over- or under-specific argument spans. Such errors may also lead to uninformative extractions, e.g. when a negation is missed. A good example on how this behaviour affects the quantitative performance of OpenIE systems can be seen by comparing the scores of the Stanford OpenIE (SIE) for the NYT-222 dataset and both matching standards (a) and (b) from figure 2.2. Comparing both strict and relaxed containment matching strategies, where relaxed is more forgiving regarding boundary errors, the strict strategy reduced the quantitative scores to a total of 0. This observation is best explained by looking at the number of boundary errors produced by the Stanford OpenIE system, noticing that it has the highest count among all systems.

2.1.3 Recommendation

Especially in the fashion domain, where training data is sparse, a data scientist in the need for a relation extraction system may tend to unsupervised systems like the aforementioned OpenIE approaches. Our analysis has shown that these approaches are not reliable enough yet for our task of extracting relations from largely unstructured data for the fashion domain. The relations extracted by OpenIE systems do suffer from a lot of errors that are specific to their nature e.g. uninformative extractions or boundary errors. Such noisy extracted relations induce a lot of erroneous information into a downstream system. This may especially affect idiosyncratic domains like texts to be found in fashion blogs and social media feeds. The helpfulness of such systems to generate training data for "classical" typed relation extractors is also limited, especially because of underspecific argument spans that make it hard to extract the correct entity pairs.

2.2 SECTOR

When searching for information, a human reader first glances over a document, spots relevant sections and then focuses on a few sentences for resolving their intention. However, the high variance of document structure complicates to identify the salient topic of a given section at a glance. To tackle this challenge, we present SECTOR, a method to support machine reading systems by segmenting documents into coherent sections and jointly assigning topic labels to each section. Our deep neural network architecture learns a latent topic embedding over the course of a document that can be leveraged to classify local topics from plain text and segment a document at topic shifts. In addition, we contribute WikiSection, a publicly available dataset with 242k labeled sections in English and German. From our extensive evaluation of 20 architectures, we report a highest score of 71.6% F1 for the segmentation and classification of 30 topics from the English city domain, scored by our SECTOR LSTM model with bloom filter embeddings and bidirectional segmentation. This is a significant improvement of 29.5 points F1 compared to state-of-the-art CNN classifiers with baseline segmentation.

2.2.1 Paragraph Classification

We introduce SECTOR, a neural embedding model that predicts a latent topic distribution for every position in a document. Based on the task shown in figure 2.4, we aim to detect M sections $\mathbf{T}_{0\ldots M}$ in a document $\mathbf{D}$ and assign topic labels $\mathbf{y}_j = \text{topic}(\mathbf{S}_j)$, where $j = 1, \ldots, M$. Because we do not know the expected number of sections, we formulate the objective of our model on sentence level and later segment based on the predictions. Therefore, we assign each sentence $\mathbf{s}_k$ a sentence topic label $\bar{\mathbf{y}}_k = \text{topic}(\mathbf{s}_k)$, where $k = 1, \ldots, N$. Thus, we aim to predict coherent sections with respect to document context:

$$p(\bar{\mathbf{y}}_1, \ldots, \bar{\mathbf{y}}_N) = \prod_{k=1}^{N} p(\bar{\mathbf{y}}_k \mid \mathbf{s}_1, \ldots, \mathbf{s}_N) \tag{2.1}$$

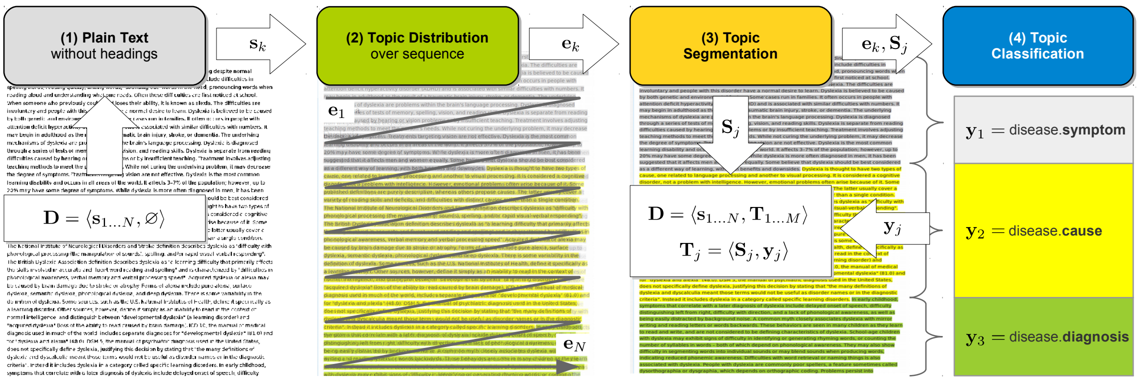

Figure 2.4: WikiSection Task: A plain text document without additional information about its structure will get structured into coherent topical segments where each segment is assigned a topic label [3].

We approach two variations of this task: for WikiSection-topics, we choose a single topic label $\mathbf{y}_j \in \mathcal{Y}$ out of a small number of normalized topic labels. However, from this simplified classification task arises an entailment problem, because topics might be hierarchically structured. For example, a section with heading *"Genetic causes"* might describe *genetics* as a subtopic of *cause* although this is not expressed explicitly. Therefore, we also approach an extended task WikiSection-headings and assign all words in the section heading

$\mathbf{z}_j \subset \mathcal{Z}$ as multi-label bag over the original heading vocabulary. This turns our problem into a ranked retrieval task with a large number of ambiguous labels, similar to [10]. It further eliminates the need for normalized topic labels. For both tasks, we aim to maximize the log likelihood of model parameters Θ on section and sentence level:

$$\begin{aligned} \mathcal{L}(\Theta) &= \sum_{j=1}^{M} \log p(\mathbf{y}_j \mid \mathbf{s}_1, \ldots, \mathbf{s}_N; \Theta) \\ \bar{\mathcal{L}}(\Theta) &= \sum_{k=1}^{N} \log p(\bar{\mathbf{y}}_k \mid \mathbf{s}_1, \ldots, \mathbf{s}_N; \Theta) \end{aligned} \tag{2.2}$$

Our Sector architecture [3] consists of four stages shown in figure 2.4: sentence encoding, topic embedding, topic classification and topic segmentation.

For a detailed explanation of each architectural component and results on various data sets we refer to our publication at TACL 2019 [3].

2.2.2 Recommendation

Semi-Supervised Learning Sector solves the important problem of missing training data by leveraging existing structures in documents, such as headlines, subheads and paragraph headings. Hence, each text that follows these characteristics can be used as training data without any further labeling. Often these texts come rather as internal reports written by experts in the company than blogs.

Robust for even tens and hundreds of classes SECTOR has been shown to be robust for many classes. That is relevant for the fashion domain as well, since relation extraction often needs to map to a fragmented schema of many different relationship types.

SECTOR recognizes relations across sentences. In contrast to nearly all other OpenIE or supervised methods, SECTOR can recognize potential relation topics even across sentence borders. That is particularly helpful for reaching a high recall, since other methods often can not capture the relation topic across sentences and therefore have a high ratio of false negatives.

SECTOR does not recognize attributes explicitly. However, SECTOR can not recognize detailed span information about a particular attribute for a particular relation. In this case, we recommend combining SECTOR with NER and NEL methods, such as our TASTY or Zalando's FLAIR.

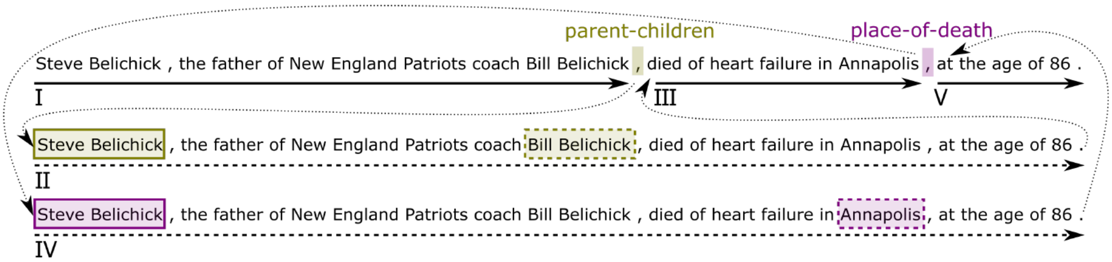

Figure 2.5: Overview of the hierarchical relation extraction process. In the first line (I) and (III) show phrases that map to relations as per the high-level policy. Below, (II) and (IV) show the low-level policy steps for respective relations. (V) depicts a phrase which doesn't map to a relation, and arrows indicate how the algorithm flows through the sequence.

2.3 Hierarchical Reinforcement Learning

This section discusses HRL-RE("A Hierarchical Framework for Relation Extraction with Reinforcement Learning" [14]). This work represents the current state of the art for relation extraction as presented at AAAI 2019. We implement and apply this model to our corpus, and evaluate it for use in the fashion domain. HRL-RE aims to improve results on the RE task by solving it together with the NER task in unison. In order to achieve this, this model employs a stacked deep learning framework which is trained using a hierarchical reinforcement learning scheme. Among other advantages this scheme allows the model to easily deal with multiple, overlapping relations in a single sentence, whereas other models often can only classify a single one. An overview for this process is depicted in figure 2.5. The Hierarchical Reinforcement Learning works in two stages, the high-level RE stage and the low-level NER stage. They each define a separate Markov Decision Process (MDP) with disjoint action-, and state-spaces and the model accordingly learns two different policies μ and π respectively in order to solve them. The policies, given a sequence of tokens $S_i = t_1, t_2, t_3, ..., t_n$ choose an action a_t^{RE} respectively a_t^{NER} for each token in the sequence. They are discussed in more detail in the following sections.

2.3.1 High-Level Policy

The High-Level policy μ is the starting policy, run when the model begins processing a sequence. This policy works somewhat differently from conventional RE in two major ways:

- It aims to not only infer the type of the relation as a sequence classification, but also the relation indicator i.e. the word or phrase that indicates whether a relation is present and of what type it is. Of note is, that this relation indicator is learned in an unsupervised fashion, no annotations are necessary

- In this step no entity annotations are necessary, as instead of the usual approach of doing NER first and then RE on the recognized entities, this approach does the reverse.

This policy can be understood as an iterative sequence classification policy. It processes the sequence token by token, in each step predicting either -1 if no relation has been found yet, or the index of the relation type it has found an indicator for. The state s_t of the underlying MDP can then be described as

$$s_t^h = f^h(W_s^h[h_t; v_t^r; s_{t-1}]) \tag{2.3}$$

with h_t being the output of a Bi-LSTM encoder at the current time step, v_t being the embedding vector of the last relation that was predicted by the high-level policy that wasn't a non-relation, and s_{t-1} the state at the last timestep. f^h is a function approximated by a small feed-forward network. As the reward of the high-level MDP the F_β score of the relations of S is used. It is described as:

$$F_\beta(S) = \frac{(1+\beta^2) * precision * recall}{\beta^2 * precision + recall}$$

Finally, the action space encompasses all relation types we aim to extract, plus the action that specifies a non-relation.

As soon as the policy predicts a value different than -1 this policy is paused, and the low-level policy takes over. Once the low-level policy finishes, the high-level policy resumes at the token where it last predicted a relation indicator until it either finds the next relation indicator, or has processed all tokens in the sequence.

2.3.2 Low-Level Policy

The low-level policy π also processes the sequence iteratively, with the important difference being, that in addition to the sequence tokens it also uses the prediction of the high-level policy as an input. Correspondingly, the state of this MDP can now be described as:

$$s_t^l = f^l(W_s^l[h_t; v_t^e; s_{t-1}; c_{t'}])$$

with c_t being the embedding vector of the relation that was chosen by the high-level policy. This simplifies the task of NER greatly. While conventional, general purpose NER models are required to learn to extract any kind of entity found in the dataset, this hierarchical model allows the neural network to learn a relation specific entity recognizer for each relation the high-level policy predicts. This Divide-and-Conquer approach reduces complexity and with that the burden on the network.

In order to extract the entities a token annotation scheme is proposed that assigns one of the following tags to each token:

- 0 : O non-entity

- 1 : S_I inside of a source entity
- 2 : T_I inside of a target entity
- 3 : O_I inside of not-concerned entity
- 4 : S_B begin of a source entity
- 5 : T_B begin of a target entity
- 6 : O_B begin of not-concerned entity

This annotation scheme distinguishes between source and target entities(for directed relations), entities that don't belong to the relation type currently selected by the high-level policy, and entities that do not represent an entity at all. The action space of the MDP corresponds to exactly these tokens. It's reward, like the reward of the high-level policy is based on a gold standard:

$$r_t^l = \lambda(y_t) * sign(a_t^l = y_t(o_{t'}))$$

where y is the the gold standard tag label, a_t is the tag generated with policy π and λ is a bias term, which reduces rewards for non entity tag (6). Once this policy has tagged all tokens with the aforementioned tags, the high-level policy resumes.

2.3.3 Model and Training Scheme

The deep neural network used in this approach consists of two components. The first is a multi-layer Bi-LSTM encoder that is initialized with GLoVe [9] vectors. This encoder transforms the token sequence into a generalized vector representation which is then used for both the RE and NER tasks. On top of that simple feed-forward networks are trained that are task specific, with one set of layers for the RE task and then separate sets of layers for each relation type that solve the NER task. In order to learn the reinforcement learning policies the REINFORCE algorithm is employed. REINFORCE is a policy gradient method, meaning it learns the policy directly, without requiring to learn value- or action-value functions. One problem which commonly arises when using reinforcement learning, particularly in the text domain, is that training a model from scratch solely on a sparse reward signal is incredibly difficult and slow. This approach solves this problem by first doing a few epochs of supervised learning, which results in an already functional model. Additionally, during supervised learning this approach employs teacher forcing. The supervised model is then used in a warm start procedure to jump-start the reinforcement learning process. This speeds up convergence dramatically.

2.3.4 API

We implement a REpresentational State Transfer (REST) Application Programming Interface (API) for HRL-RE that we provide for use in further deliverables. This API exposes both the NER, and the RE capabilities of this model. It currently expects a single sentence in string form and returns the predicted relation type and entities in JavaScript Object Notation (JSON) format. The JSON schema we use follows the TeXoo JSON schema. TeXoo is an open source Natural Language Processing (NLP) framework which comes with it's own document model with annotation schemes for different NLP tasks. Figure 2.6 shows an example of the JSON format.

```
{
  "begin": 1,
  "length": 128,
  "sentences": [{
    "begin": 1,
    "end": 128,
    "text": "Steve Belichick, the father of New England Patriots coach Bill
    Belichick, died of heart failure in Annapolis, at the age of 86."
  }],
  "annotations": [{
    "class": "Relation",
    "type": {
      "class": "RelationType",
      "name": "DIED-IN"
    },
    "source": {
      "class": "RelationAnnotation",
      "begin": 1,
      "length": 15,
      "text": "Steve Belichick"
    },
    "target": {
      "class": "RelationAnnotation",
      "begin": 100,
      "length": 9,
      "text": "Annapolis"
    }
  }]
}
```

Figure 2.6: Example of a relation in the TeXoo JSON format.

2.3.5 Recommendation

Table 2.2 shows results from our datasets from Section 1. We now discuss if this approach can be applied to FashionBrain.

Only four classes reach notable F1 above 60% and seven classes between 40% and 60% In general, our results confirm the state-of-the-art: Relation extraction is an unsolved problem, F1 measures are around 60% for frequent classes and much lower for infrequent or idiosyncratic classes.

Higher F1 scores correlate with more labeled training data Only two classes reach a notable F1 score above 40% and have less than 100 training examples. Overall we note a drastic reduction of F1 if only a few hundred or even fewer training examples are available. This means that for FashionBrain we need methods for labeling sufficient training data in particular for 'rare' relations with few efforts.

In addition, semi-supervised methods, such as SECTOR, can compensate the lack of training data by extracting information from already labeled data, such as headlines and correlated text.

Precision is king. Overall, precision is often much higher than recall. We assume that the extractor therefore can not generalize well to unseen data but can at least establish partially robust filters for false positives. However, the insufficient generalization capabilities lead the extractor to ignoring many false negatives. This holds true in particular when only a few examples are given at training time.

Binary relations only and not across sentence borders. Our state-of-the-art extractor is not capable of extracting n-ary relationship types. However, often database schemes that may serve as an endpoint for relation extraction and are used for trend analysis, require n-ary relations. Worse, recall is drastically limited to the existence of relation attributes within a single sentence. This is often not the case in practice. Rather, arguments in relations are represented as pronouns, anaphoras and other syntax structures, but not as extractable and linkable entities, that can be captured by NER/NEL frameworks, such as TASTY or FLAIR.

Overall, we conclude that we should further invest in efficient schemes for labeling training data for relation extraction. Furthermore, approaches like SECTOR, can compensate the inability of supervised approaches and can detect relation topics even across sentences.

Table 2.2: Relation Classes in Combination Corpus

Relation	Count	Precision	Recall	F1
/location/location/contains	4453	0.76	0.46	0.57
/location/administrative_division/country	1469	0.86	0.52	0.65
/location/country/administrative_divisions	1469	0.86	0.52	0.65
/location/country/capital	1443	0.83	0.53	0.65
/people/person/nationality	840	0.69	0.56	0.62
/people/person/place_lived	582	0.66	0.42	0.51
/person/title/title	500	0.09	0.09	0.09
/business/person/company	421	0.71	0.65	0.67
/organization/person/top_members/employees	346	0.38	0.38	0.38
Component-Whole	309	0.34	0.34	0.34
/people/person/place_of_birth	267	0.68	0.39	0.5
/person/organization/employee_of	263	0.04	0.04	0.04
Message-Topic	261	0.22	0.22	0.22
Entity-Origin	255	0.41	0.4	0.4
Member-Collection	229	0.44	0.44	0.44
Product-Producer	223	0.28	0.28	0.28
/organization/organization/alternate_names	211	0.17	0.17	0.17
Content-Container	191	0.49	0.49	0.49
/business/company/founders	185	0.6	0.49	0.54

/person/number/age	168	0.25	0.24	0.25
/person/city/cities_of_residence	163	0.04	0.04	0.04
/people/ethnicity/geographic_distribution	155	0.59	0.48	0.53
/person/country/countries_of_residence	134	0.04	0.04	0.04
/location/neighborhood/neighborhood_of	133	0.51	0.24	0.32
/people/deceased_person/place_of_death	118	0.63	0.32	0.42
/person/nationality/origin	107	0.02	0.02	0.02
/organization/country/country_of_headquarters	105	0.09	0.09	0.09
/person/criminal_charge/charges	103	0.05	0.05	0.05
/business/company/major_shareholders	90	0.87	0.84	0.85
/person/person/parents	88	0.0	0.0	0
/location/us_state/capital	86	0.54	0.25	0.34
/organization/city/city_of_headquarters	77	0.11	0.12	0.12
/person/stateorprovinces_of_residence	73	0.03	0.03	0.03
/organization/person/founded_by	68	0.01	0.01	0.01
/person/person/spouse	66	0.03	0.03	0.03
/person/person/other_family	60	0.0	0.0	0
/organization/organization/parents	59	0.02	0.02	0.02
/person/person/siblings	55	0.0	0.0	0
/person/date/date_of_death	54	0.04	0.04	0.04
/person/cause_of_death/cause_of_death	52	0.0	0.0	0
/person/religion/religion	47	0.02	0.02	0.02
/organization/organization/subsidiaries	43	0.0	0.0	0
/location/us_county/county_seat	41	0.63	0.31	0.41
/organization/date/founded	37	0.1	0.11	0.1
/person/person/children	37	0.03	0.03	0.03
/people/place_of_interment/interred_here	34	0.8	0.22	0.34
/people/deceased_person/place_of_burial	34	0.8	0.22	0.34
/location/province/capital	34	0.59	0.2	0.29
/people/person/children	33	0.36	0.23	0.28
/person/duration/age	32	0.0	0.0	0
/person/organization/schools_attended	30	0.07	0.07	0.07
/organization/url/website	26	0.19	0.19	0.19
/person/city/city_of_death	26	0.0	0.0	0
/organization/organization/members	26	0.0	0.0	0
/person/location/cities_of_residence	26	0.0	0.0	0
/person/country/origin	25	0.0	0.0	0
/business/company/place_founded	24	0.29	0.19	0.23
/organization/number_of_employees	19	0.06	0.05	0.05
/people/person/ethnicity	14	0.31	0.2	0.24
/organization/organization/member_of	14	0.0	0.0	0
/person/nationality/countries_of_residence	13	0.0	0.0	0

/person/stateorprovince_of_death	12	0.0	0.0	0
/sports/sports_team/location	10	0.5	0.5	0.5
/person/person/alternate_names	10	0.0	0.0	0
/person/country/country_of_death	9	0.0	0.0	0
/person/date/date_of_birth	9	0.0	0.0	0
/person/stateorprovince_of_birth	8	0.0	0.0	0
/person/location/stateorprovinces_of_residence	8	0.0	0.0	0
/organization/person/shareholders	7	0.0	0.0	0
/people/person/religion	6	0.5	0.38	0.43
/organization/organization/shareholders	6	0.0	0.0	0
/organization/location/city_of_headquarters	5	0.0	0.0	0
/person/city/city_of_birth	5	0.0	0.0	0
/organization/country/members	5	0.0	0.0	0
/person/country/country_of_birth	4	0.0	0.0	0
/location/br_state/capital	4	0.5	0.25	0.33
/time/event/locations	4	0.0	0.0	0
/film/film/featured_film_locations	4	0.0	0.0	0
/film/film_location/featured_in_films	4	0.0	0.0	0
/organization/country/member_of	3	0.0	0.0	0
/organization/location/country_of_headquarters	3	0.0	0.0	0
/organization/misc/alternate_names	2	0.0	0.0	0
/organization/country/parents	2	0.0	0.0	0
/person/location/stateorprovince_of_death	2	0.0	0.0	0
/person/location/city_of_death	2	0.0	0.0	0
/organization/location/subsidiaries	1	0.0	0.0	0
/person/misc/alternate_names	1	0.0	0.0	0
/person/location/employee_of	1	0.0	0.0	0
/organization/location/parents	1	0.0	0.0	0
/person/location/countries_of_residence	1	0.0	0.0	0
/person/nationality/country_of_birth	1	0.0	0.0	0

3 Discussion

Our thoughtful and deep inspection of three approaches to relation extraction in the fashion domain confirmed state-of-the-art results: Relation extraction remains a hard and unsolved problem, reaching often F1 scores below of 60%. Further, state-of-the-art leveraging sophisticated neural network technologies, such as hierarchical reinforcement learning, cannot overcome the problem of sparse training data and therefore still generalize poor. In addition, these approaches only provide capabilities for binary relations and lack the ability of n-ary relation extraction and extraction across sentence boundaries.

We were also able to observe that four leading OpenIE systems are not applicable at all to the problem of relation extraction in FashionBrain. First, these systems depend on additional syntactic-lexical patterns that are often hard-coded and do not match the language of neither other test datasets nor of the fashion domain. Our careful analysis unraveled that each OpenIE system is rather over-fitted on the single dataset it was designed for.

Therefore, we even designed a novel approach called SECTOR that provides a new paradigm. Instead of extracting relations and extract spans describing attributes for relations, SECTOR extracts sentences or paragraphs and assigns a distribution of latent or discrete relation topics. We will investigate this promising paradigm further.

3.1 Improving the Model of HRL-RE

Significant investments must be made for improving models. This is not only a problem for FashionBrain, but rather for an entire research community in artificial intelligence and computational linguistics. With respect to FashionBrain, we envision a few ways how our results could eventually be improved:

(1) The HRL-RE model, which lends itself well to extracting multiple overlapping relations in a sentence, could be modified to solve the difficult task of n-ary relations at least for some cases.

(2) In cases where multiple relations fuzzily share a source- or a target entity, which is often the case for example with enumerations, binary relations a-b and b-c can be combined to a compound n-ary relation a-b-c. This can be done with simple heuristics and without the use of deep learning methods.

(3) In order to improve the results of HRL-RE one straightforward method is to replace the untrained LSTM encoders with a strong, pre-trained, contextualized

baseline. For example, using BERT or GPT-2 as a drop-in replacement should improve transfer learning capabilities. These models, through their large representative power and special pre-training procedure, manage to capture world knowledge and global context much more readily than the small LSTM based model used in this deliverable. Because of this inbuilt knowledge, they should prove much better at handling especially the long tail of classes with only few examples.

(4) Our model, in particular in conjunction with Heideltime [13], a temporal sequence tagger, could prove useful for extracting timestamps from texts, which can then be used as time series data in the model developed in D5.3.

3.2 Curating Own Fashion Training Data – Fashion Communication Corpus

As a second approach, we are working together with University of Sheffield on building a new dataset which can then be used in deliverable D4.4 to test the reported techniques further. In this approach we leverage an as yet unlabeled dataset in the Fashion Communication Corpus of Hong Kong Polytechnic University (HKFCC) and utilize crowd workers to annotate both the entities and the relation types. The HKFCC contains 1 million words of unlabeled documents from multiple categories including news articles, blog posts, blog comments, research articles, styling tips and product launches.

To access the corpus, we can query sample sentences through the use of the "Simple Query Language" [8, Chapter 6]. First we manually evaluated the different document categories and came to the conclusion that blog posts and comments are not usable for our case, since their language is very specific and different from our target domains. Also, they only very rarely contain relevant relations. In contrast, news reports and research articles are largely useful, with a comparatively high percentage of sentences per document containing an annotatable relation.

For these categories we then formulate queries that extract sentences for further annotation and filter out most of the irrelevant sentences. We focus here on seven relation types that we want the crowd workers to annotate:

1. PERSON wears/uses PRODUCT
2. PERSON attends EVENT
3. COMPANY sponsors EVENT
4. COMPANY releases/makes/produces/designs PRODUCT
5. PERSON desires/wants/is excited for PRODUCT
6. PERSON wins AWARD
7. PRODUCT consists of/is made of COMPONENTS

Both the relations and suitable patterns pertaining to those relations were selected after doing a qualitative analysis of fashion news sources (e.g. vogue.co.uk, glamour.com, fashionunited.uk). We focus on these relations, as they are relation types most commonly found in the fashion media we evaluated, and using only relatively few relation types makes it easier for the crowd workers to label the samples correctly, resulting in better data. In the analysis we identified the most common keywords pertaining to the aforementioned relations and used them to construct seed patterns to query a first batch of samples from HKFCC. When constructing Simple Query Language patterns from these keywords we focused on very simple patterns using only the keywords themselves, Part of Speech tags (denoting whether a word is a noun, verb, etc.) and wildcards. This keeps the patterns as broad as possible thereby producing the most diverse set of samples. After a manual evaluation of the samples matched by those initial patterns regarding their number, syntactic quality, and relevance to the relations, we expanded them by adding semantically similar verbs. We continued this iterative process until additional patterns did not yield a significant amount of relevant samples. The final Simple Query Language [8] query used to retrieve relevant sentences is shown in Listing 3.1.

```
(
dress*_V* | suit*_*V | cloth*_V | wear*_V* | wore_V* |
worn_V* | attend*_V* | join*_*V | event_N* | invite*_V* |
love*_V* | like*_V* | want*_V* | desire*_V* | cho*se*_V* |
w?n_V* | achieve*_V* | award_N* | release*_V* |
launch*_V* | introduce*_V* | present*_V* | tour*
)
```

Listing 3.1: Query used to retrieve sentences from the HKFCC-corpus (_V) selects only verbs, (_N) selects only Nouns, () matches any number of any character and (—) denotes an OR between different subqueries.

3.2.1 Crowdsourced Annotations for the Fashion Communication Corpus

Query

We queried the HKFCC using the query set described in Listing 3.1, grouping the queries by topic (e.g. for relation 7 we use the query `ove*_V* | like*_V* | want*_V* | desire*_V* | cho*se*_V*`) and by source (blog, news, press release etc.) obtaining a dataset of 7350 text fragments, each containing a sentence with up to 50 words on the left and on the right of the core verb.

Pre-process

After this phase, we performed a heuristic text pre-processing to removing duplicates, and to select the sentence boundaries containing the core verb. It is important to note that, while each query has a high probability to extract the desired relation, we can have: fragments in which the relation is not present and fragments in which multiple relations are present. Moreover, in order to obtain a dataset useful for training, it is necessary to identify the boundaries of the different entities in the text fragments. To do that, we will make use of crowdsourcing, as described in the next section.

Crowdsourcing Task

Each sentence needs to be annotated by the crowd. The annotation phase is not trivial, because it is necessary to identify the relations in a sentence, and for each relation to locate the portions of the text corresponding to the different entities: for example for the sentence: "Madonna wears nike shoes", we need the annotator to identify relation 1, and to highlight the text "Madonna" for the `PERSON` entity, and "nike shoes" for the `PRODUCT` entity. Moreover, sometimes one of the two entities might be missing.

We developed a custom crowdsourcing interface to allow crowdworkers to easily annotate sentences according to the described structure. We trained them in a pilot phase and selected workers that had an accuracy higher than 70%.

Some screenshots of the crowdsourcing interface is shown in Figures 3.1-3.3, while the instructions are shown in Figure 3.4.

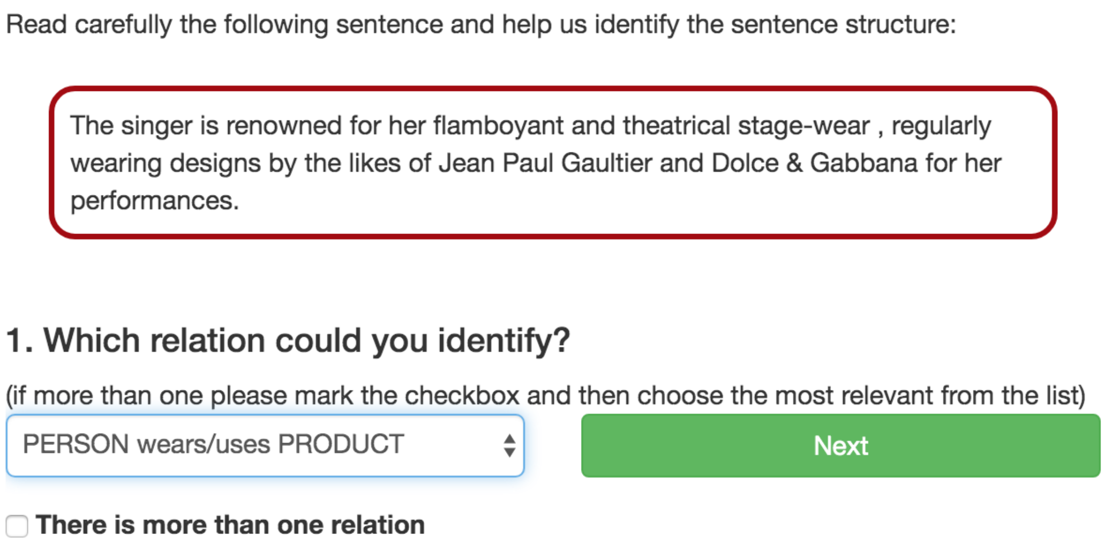

Figure 3.1: Fashion Communication Corpus of Hong Kong Polytechnic University annotation task. Relation recognition step.

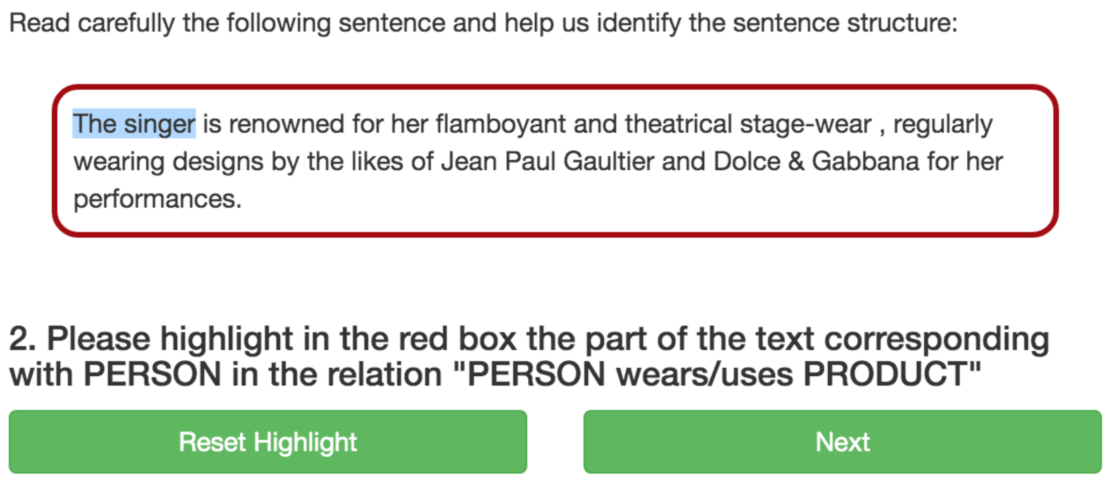

Figure 3.2: Fashion Communication Corpus of Hong Kong Polytechnic University annotation task. Entity recognition step (highlighting `PERSON`).

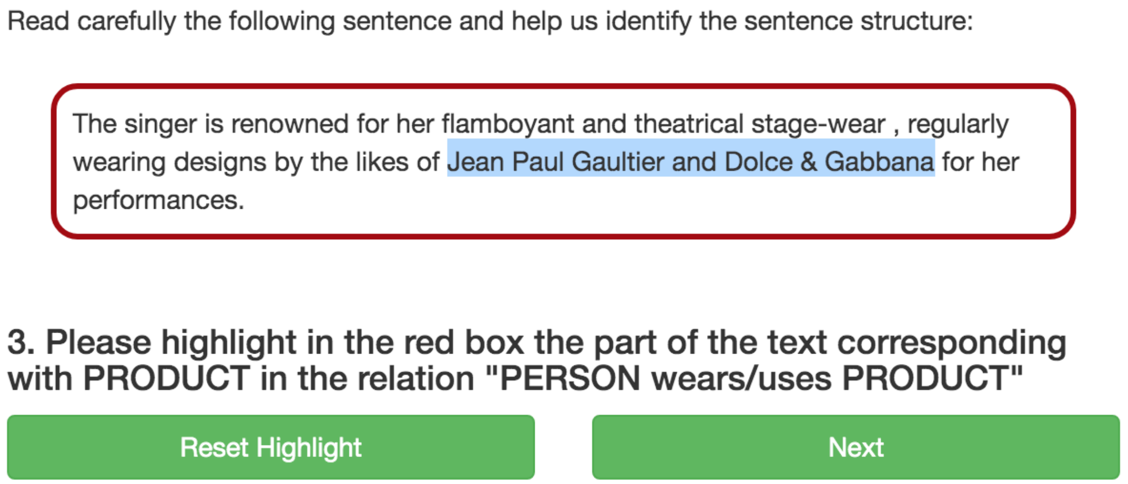

Figure 3.3: Fashion Communication Corpus of Hong Kong Polytechnic University annotation task. Entity recognition step (highlighting `PRODUCT`).

Overview

In this task we will ask you to classify the text in the red box. It is useful to use to train our algorithms to recognise fashion text.

Steps

The task is divided into three parts:

1. Find the fashion relation. For example select "PERSON wears/uses PRODUCT" if the text is about someone wearing or using a fashion item, like "Madonna [...] was sporting some Nike red shoes. If the relation is not on the list select "Relation Missing".
2. Highlight with the mouse the part of the text corresponding to PERSON. For example, you would highlight the word "Madonna" in this example. If PERSON is missing or is it not clear who is it, you can leave it empty.
3. Highlight the part of the text corresponding to PRODUCT. For example, you would highlight the words "Nike red shoes" in this example. If PRODUCT is missing, you can leave it empty.

There could be more than one relations, for example "I'm wearing the Piarry Boot by ZigiNY in tan." contains both the relation "wearing" (I wear Piarry Boot) but also the relation "design" (ZigiNY designs Piarry Boot). In that case please select the most important one and click the checkbox "there are multiple relations".

Identifying the right relation

This is the list of relations (with examples):

- PERSON wears/uses PRODUCT. For example, [Mrs. Hilton] chose a [Chanel bag] (note that the verb could be different)
- PERSON attends EVENT. For example [Fashion Week] expects many important guests like [Heidi Klum] (note that here the first element of the relation appears at the end)
- COMPANY sponsors EVENT. For example [Nike] is presenting their new collection at [Runbase] (note that the verb is different, it is still the same relation)
- PERSON/COMPANY releases/makes/produces/designs PRODUCT. For example, [Adidas] releases the new line of [predator football shoes] (note that you can have multiple words)
- PERSON desires/wants/is excited for/likes PRODUCT. For example, [Tom] says he is obsessed with [kimono jackets] (note that you can have multiple words and the verb can be different)
- PERSON wins AWARD. For example [Sandy Powell] accepts the [BAFTA] in costume design.
- PRODUCT consists of/is made of COMPONENTS. For example, The [dress] consists of a [black non transparent body]

Figure 3.4: Fashion Communication Corpus of Hong Kong Polytechnic University annotation task. Instructions.

4 Conclusions

This report demonstrates and evaluates 3 different approaches to relation extraction in the fashion domain. We give the following recommendations on their use: OpenIE, while resulting in the most noisy predictions, is the only possible option to work directly on fashion data when a fashion dataset is not available. SECTOR can yield strong results in cases where relation types are consistent across paragraphs and the sequences relations are extracted from are sufficiently long. HRL-RE yields the strongest results, provided a large enough fashion relation extraction dataset is available and should be used in such cases. This report also discusses our two approaches to coping with the unavailability of a fashion dataset.

First, we propose to use transfer learning on a dataset specifically designed to fit the target domain and create such a dataset for fashion. Secondly, we collaborate with University of Sheffield to crowdsource a fashion-themed relation extraction dataset: we describe the process of fashion relation collection from the Fashion Communication Corpus of Hong Kong Polytechnic University, and the ongoing process of annotation through crowdsourcing. When annotated, this dataset will allow to extend the analysis conducted in this deliverable on a real-world fashion dataset. We will include this extended analysis in D4.4.

Bibliography

[1] Alan Akbik and Alexander Löser. Kraken: N-ary facts in open information extraction. In *Proceedings of the Joint Workshop on Automatic Knowledge Base Construction and Web-Scale Knowledge Extraction*, pages 52–56. Association for Computational Linguistics, 2012. 00000.

[2] Gabor Angeli, Melvin Jose Johnson Premkumar, and Christopher D. Manning. Leveraging linguistic structure for open domain information extraction. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 344–354, 2015. 00143.

[3] 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.

[4] Michele Banko, Michael J. Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open Information Extraction from the Web. In *IJCAI*, volume 7, pages 2670–2676, 2007. 00000.

[5] Luciano Del Corro and Rainer Gemulla. Clausie: Clause-based open information extraction. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 355–366. International World Wide Web Conferences Steering Committee, 2013. 00000.

[6] Anthony Fader, Stephen Soderland, and Oren Etzioni. Identifying relations for open information extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545. Association for Computational Linguistics, 2011. 00000.

[7] Filipe Mesquita, Jordan Schmidek, and Denilson Barbosa. Effectiveness and Efficiency of Open Relation Extraction. In *EMNLP'13*, pages 447–457. Association for Computational Linguistics, 2013. 00028.

[8] Matthew Brook O'Donnell. Sebastian hoffmann, stefan evert, nicholas smith, david lee and ylva berglund prytz, corpus linguistics with bncweb – a practical guide (english corpus linguistics 6). frankfurt am main: Peter lang, 2008. pp. xii 288. - wendy anderson and john corbett, exploring english with online corpora: An introduction. basingstoke: Palgrave macmillan, 2009. pp. xiii 205. *English Language and Linguistics*, 15(3):558–564, 2011. doi: 10.1017/S1360674311000177.

[9] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural

Language Processing (EMNLP), pages 1532–1543, 2014. URL http://www.aclweb.org/anthology/D14-1162.

[10] Yashoteja Prabhu and Manik Varma. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 263–272, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2956-9. doi: 10.1145/2623330.2623651. URL http://doi.acm.org/10.1145/2623330.2623651.

[11] 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.

[12] Gabriel Stanovsky and Ido Dagan. Creating a Large Benchmark for Open Information Extraction. In *EMNLP'16*, page (to appear), Austin, Texas, 2016. ACL. 00001.

[13] Jannik Strötgen and Michael Gertz. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298, 2013. doi: 10.1007/s10579-012-9179-y.

[14] 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.

[15] Aaron Steven White, Drew Reisinger, Keisuke Sakaguchi, Tim Vieira, Sheng Zhang, Rachel Rudinger, Kyle Rawlins, and Benjamin Van Durme. Universal Decompositional Semantics on Universal Dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016. 00004.

[16] Ying Xu, Mi-Young Kim, Kevin Quinn, Randy Goebel, and Denilson Barbosa. Open Information Extraction with Tree Kernels. In *HLT-NAACL*, pages 868–877, 2013. 00029.