Horizon 2020

fashion BRAIN project

Understanding Europe's Fashion Data Universe

Demo on Textual Image Search

Deliverable number: D6.5

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	Demonstrator (D)
Dissemination level	**Public** (P)
Contractual Delivery Date	31/09/2019
Number of pages	21, the last one being no. 15
Authors	Alan Akbik - Zalando
Peer review	Alessandro Checco, Kathryn MacKellar - USFD

Change Log

Version	Date	Status	Partner	Remarks
0.1	15/09/2019	Draft	Zalando	
1.0	25/09/2019	Final	Zalando	

Deliverable Description

This deliverable consists of a image search prototype system which uses all of the data collected and allows users to search by images, collects user feedback and is able to periodically improve its results based on this interaction data. It extends the textual component of D6.3 by NLP and multi-linguality primarily targeting on German, English, French, and Italian.

Abstract

With deliverable D6.5, we present a fully functioning neural information retrieval system that showcases the text-to-image search capabilities developed over the course of work package 6. The system is capable of retrieving product images based on descriptive textual queries. For instance, the query “rotes Kleid mit weißen Streifen” (engl. “red dress with white stripes”) should retrieve a ranked list of product images that match this description. This deliverable connects the output of several work streams in FashionBrain: (1) The neural two-tower architecture presented in deliverable D6.3, created to address the shortcomings of traditional search approaches (described in D1.2). (2) FEIDEGGER, the annotated dataset described in deliverables D6.1 and D6.4, used as *training data* for the neural information retrieval approach. And (3) FLAIR, our open source text representation framework described in D6.2, used to improve textual query understanding in the system. This demonstration report describes how we integrate these three components to create a powerful neural search architecture and presents the results of a comparative evaluation that illustrates the impact of the text understanding module. We also illustrate how we leverage the open source framework FLAIR to showcase this technology and give an overview of the final demonstration.

A demonstrator is available online at https://fashionbrain-project.eu/demo-on-textual-image-search/.

Table of Contents

List of Figures

List of Tables

List of Acronyms and Abbreviations

NLP	Natural Language Processing
SGD	Stochastic Gradient Descent

1 Introduction

One of the core innovations of FashionBrain is to develop new information retrieval technologies to connect customers to fashion items in an online shop via full text search. For instance, the search query "white shirt with brown buttons" should yield a ranked list of matching items in the product catalogue. Whereas traditional information retrieval approaches are limited in terms of semantic coverage[1], we seek to enable customers to search for products with arbitrarily complex textual search queries that ultimately approach a form of colloquial interaction with our shop. Importantly, full text queries impose no restrictions, allowing a customer to issue any query he/she may find appropriate, including vague, exploratory and subjective queries such as "a colorful but elegant dress that I might wear at a dinner reception". Developing technologies to handle arbitrarily complex search queries will enable us to connect more customers to products and thus increase revenue of fashion e-commerce companies.

The demonstrator presented in this report is a fully functioning neural information retrieval system that showcases the text-to-image search capabilities developed in work package 6. An early version of this system was presented earlier in deliverable D6.3 in which we gave an overview of the basic "two-tower" architecture of the system. Briefly, our approach is trained to learn representations of both text and images as vectors in a shared, high-dimensional vector space. In this space, text and images with similar semantics are embedded close to each other thus allowing us to compute the similarity of text-image pairs using the cosine distance of their respective vectors. We refer the reader to the deliverable report of D6.3 for a detailed overview of the base architecture.

This deliverable describes the final system that extends the earlier prototype with a number of core novelties:

Novel text representations. Most importantly, we integrate our research on learning meaningful textual representations from very large collections of text data in many languages such as German, English, French and Italian. As illustrated in deliverable D6.2, we proposed a character-level language modeling approach to derive these representations which we refer to as "contextual string embeddings". We showed that these representations are highly meaningful and presented new state-of-the-art evaluation numbers across a whole family of classic natural language processing (NLP) tasks. With this deliverable report, we show how integrating these representations as a *query understanding module* into our neural information retrieval system significantly improves overall system performance.

[1] See deliverable D1.2 for a discussion on the limitations of traditional search approaches.

Complex textual descriptions. The prototype presented in deliverable D6.3 was trained with paired text-image data that was problematic for two reasons. First, it was impoverished in the sense that training data only contained short keyword-based search strings, thus limiting the overall semantics our neural information retrieval approach could learn to model. Second, we only used data that was proprietary to Zalando, limiting the ability of the FashionBrain consortium or third parties to reproduce our results or use our system. With this deliverable, we instead integrate the FEIDEGGER dataset created in D6.1 and D6.4 which contains rich textual descriptions obtained by crowdsourcing and has been made publicly available.

Model-based similarity. Our early prototype used cosine distance between embedding vectors to determine the overall similarity of a text and an image. This limits the notion of similarity to one of *overall similarity*: If two vectors point to a similar direction in the vector space they are considered similar. One effect of this is that the shared vector space will be learned in such a way that one area of the vector space is occupied by dresses while another is occupied by shoes and so on. This becomes problematic for queries that need to match multiple modes in the shared vector space. For instance, a query such as “wedding” would need to match both “wedding dresses” and “wedding shoes”, which with a purely cosine-based approach would reside in different corners of the vector space. We instead propose a model-based approach capable of performing linear transformations on the space of representations to address such multi-mode queries.

Taken together, these innovations enable our system to capture a much broader range of textual semantics and match them to images.

1.1 Placement of this Deliverable within FashionBrain

In this deliverable, *D6.5 “Demo on textual image search”*, we present the final neural information retrieval system developed over the course of FashionBrain. It builds on the basic architecture of D6.3, integrates the text understanding component of D6.2 and is trained using the dataset created in D6.1 and D6.4.

As such, the requirements for this demonstrator are derived from the business scenarios identified in *D1.2 “Requirement analysis document”*, in particular:

- Scenario 1: End-To-End Multi-language Search
    - Challenge 1: Mapping Search Intentions to Product Attributes
    - Challenge 2: End-To-End Learning

2 Two-Tower Embedding Architecture

As discussed in deliverable D6.3, our approach is based on a so-called “two-tower” architecture in which each tower is responsible for embedding one modality into a shared embedding space. The first tower is responsible for embedding text, while the second is responsible for embedding images. We train this two-tower architecture so that the vector representation of a text and the vector representation of an image are nearby in the shared space if they share the same semantics. Similarly, we require representations to be far apart if a text does not match an image. This process is illustrated in Figure 2.1 for a textual description, a matching image and a non-matching image (note that both the matching and the non-matching image go through the same tower, so it is a two-tower architecture even though the image shows three). Since this approach learns a way to represent arbitrary text and images in form of a meaningful vector, the process of embedding a modality in a vector space may be thought of as a form of text or image understanding.

As Figure 2.1 shows, this architecture requires the following components: A “**text model**” (or “text understanding model”) to derive a meaningful vector representation for text. An “**image model**” (or “image understanding model”) to derive a meaningful vector representation for images. A “**similarity function**” to measure the similarity between two vector representations. And a “loss function” that is used to train the architecture end-to-end given supervision by paired image-text data. We discuss our solutions to all components in turn and quantify the impact of different solutions on downstream task performance.

Experimental setup. All numbers presented in the following were obtained by training and evaluating models using the FEIDEGGER [10] dataset. We choose this dataset because it contains rich textual descriptions of fashion articles (dresses) and because we have made it publicly available. This allows third parties to easily reproduce our results and, most importantly, enables other researchers or companies to compare their neural information retrieval approaches against the one presented here. We train our neural architecture using stochastic gradient descent (SGD) with a hinge loss function. We use the standard training method of mini-batching and gradually reduce the learning rate during training with an annealing scheme against the training loss. We continue training until validation loss no longer improves and the learning rate has fully annealed.

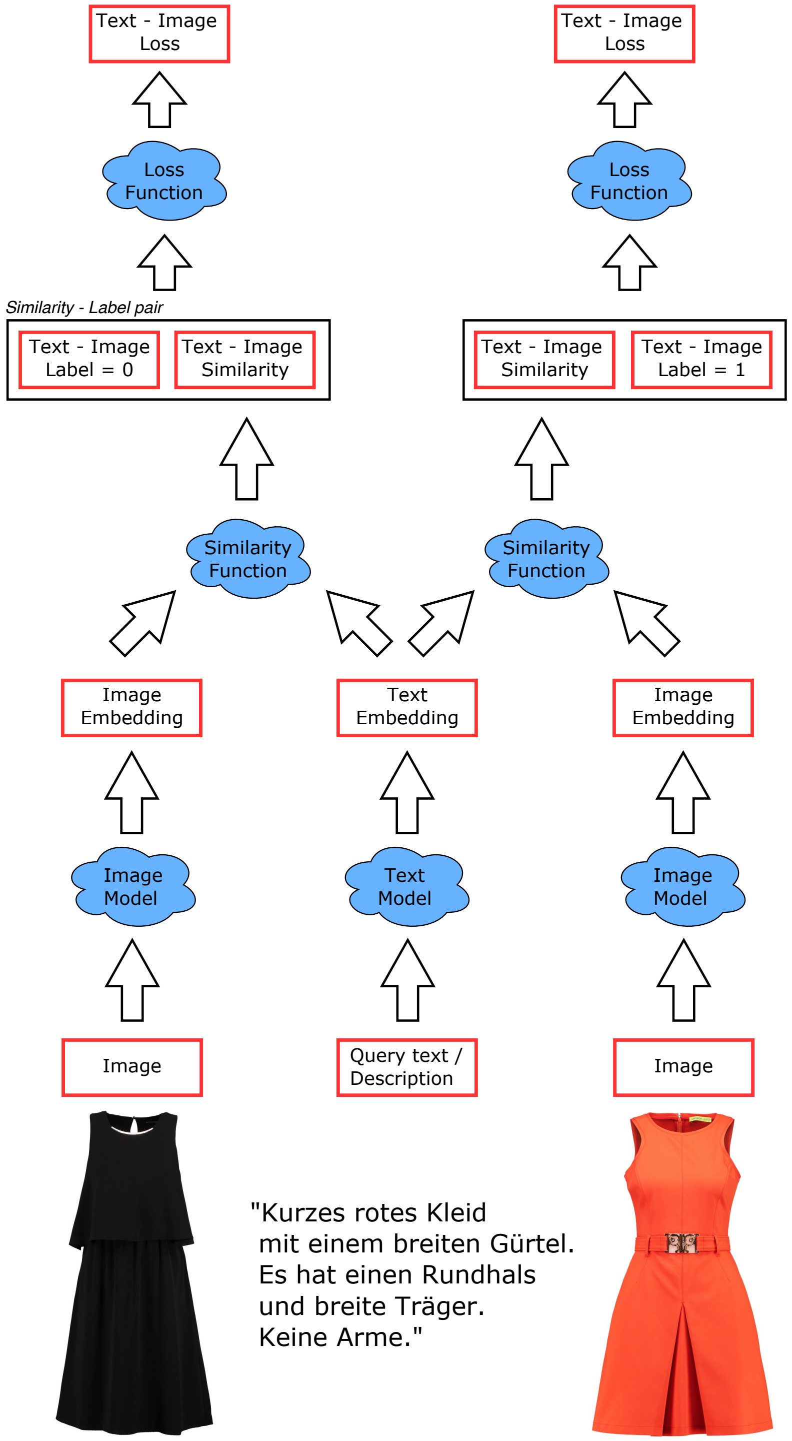


Figure 2.1: Illustration of the training procedure of our two-tower architecture. A text (middle column) is paired with a matching image (right column) and a non-matching image (left column). The objective is to embed text and images in such a way that matching image ends up close and non-matching image far apart in the final vector space.

2.1 Text Model

How to best embed text into a vector space has been an active area of research in NLP for many years. All current approaches use some form of *transfer learning* to derive meaningful representations from text. The core problem is that paired image-text data is not readily available and often needs to be manually produced. For instance, in FashionBrain we employed a costly crowdsourcing process to generate the 40.000 paired text-image data points of FEIDEGGER. But while suitable paired image-text data is rare, unlabeled plain text data is available in near-limitless quantities. The basic idea of transfer learning is to leverage unlabeled plain text data to first *pre-train* powerful text representation models that are then further fine-tuned with text-image data.

A number of approaches to pre-train word embeddings have been proposed over the past decade. Classic word embeddings are trained on the word-level by learning to predict co-occurring words [11, 12]. More recent approaches break down words into components such as syllables or individual letters to learn representations that are more robust to spelling errors or morphologically rich languages. Examples include FASTTEXT [9] and BPE [9] embeddings. Recent work has also investigated the notion of inherently **multilingual** embeddings such as MUSE [6] or multilingual Flair embeddings [3]. Most importantly, current research has moved away from the "one word, one embedding" paradigm, leading to methods that assign different vectors to the same word if it is used in a different context. For instance, the word "Tom" might be a name ("Tom Sawyer") or a fashion brand ("Tom Taylor") depending on the context in which it is used. Current approaches such as BERT [7] or the Flair embeddings [1] developed in FashionBrain are capable of modeling contextualized semantics and have shown to be very powerful for many NLP tasks.

text embedding	median rank	recall@1	recall@5	recall@10	recall@20
MUSE	6	0.198	0.477	0.618	0.749
BPE	4	0.29	0.599	0.725	0.826
Character	3	0.28	0.601	0.735	0.844
FastText	3	0.293	0.607	0.742	0.843
Flair	3	0.317	0.645	0.775	0.868
Flair-Fashion	3	**0.339**	**0.661**	**0.783**	**0.873**

Table 2.1: Impact of different text embedding models on retrieval performance.

Impact of text understanding module. We conduct an ablation study to assess the impact of different types of word embeddings on downstream task performance. In this study, the rest of the architecture is fixed as follows: A gated recurrent unit [5] (GRU) is used to combine word embeddings into a single vector representing a textual query, images are embedded with "Fashion DNA" [4] and we use a model-based similarity to match text to images. The results of this study are shown in Table 2.1.

As Table 2.1 illustrates, the choice of text model has a significant impact on retrieval quality. A Flair model trained specifically on the domain of fashion ("Flair-Fashion") outperforms all other embedding approaches by a significant margin. In particular, we note a stark improvement of all recall measures compared to non-contextualized embeddings (e.g. FastText, MUSE, etc). We also note a significant improvement over the earlier, purely character-based approach presented in deliverable D6.3 (indicated as "Character" in the table). We conclude that our pre-training approach effectively improves the neural information extraction system.

2.2 Image Model

We investigated several options for embedding images that are broadly distinguished on whether they are pre-trained or task-trained. Pre-trained models are similar to pre-trained word embeddings in that they are pre-computed over very large unlabeled image collections and so are already capable of capturing general image features. Task-trained models on the other hand are not pre-computed, but rather fully trained from scratch only with the limited text-image data available for the task at hand.

We experiment with two pre-trained models: (1) A proprietary fashion-specific solution known as "Fashion DNA" [4] and (2) a pre-trained deep-residual convolution neural network (ResNet-50) [8] that is publicly available. We also experiment with a novel architecture that we train from scratch in which we combine convolutional layers with transformer-based self attention. We refer to this approach as "TransformerConvNet". Evaluation results are presented in Table 2.2.

image embedding	median rank	recall@1	recall@5	recall@10	recall@20
ResNet-50	8	0.138	0.404	0.555	0.704
TransformerConvNet	?	?	?	?	?
Fashion DNA	3	0.33	0.6611	0.7813	0.87

Table 2.2: Impact of different image embedding models on retrieval performance.

We note that X is better than Y.

2.3 Similarity Function

We experiment with two measurements of similarity. The first is a simple cosine distance of the respective embedding vectors which we employed in our early prototype (see deliverable D6.3): So for a product image I, the network maps to a vector $\mathbf{h}_i = g(I)$. Given $\mathbf{h}_t$ and $\mathbf{h}_i$ we may compare the vectors and formulate an objective function which forces images and text which match to be "close" and otherwise "far" from one-another. To measure closeness we use cosine-similarity:

$$s(\mathbf{h}_t, \mathbf{h}_i) = \cos(\mathbf{h}_t, \mathbf{h}_i) = \frac{\mathbf{h}_t^\top \mathbf{h}_i}{\|\mathbf{h}_t\|_2 \|\mathbf{h}_i\|_2}$$

The second is a model-based similarity in which the text representation $\mathbf{h}_t$ is mapped to a image classifier $\mathbf{w}_t$ via the learned model parameters $\mathbf{A}$ and $\mathbf{b}$. The score is then obtained by applying the image classifier on image $\mathbf{h}_i$:

$$s(\mathbf{h}_t, \mathbf{h}_i) = \sigma(\mathbf{w}_t^\top \mathbf{h}_i + b_t), \quad \text{where } \mathbf{w}_t = \mathbf{A}\mathbf{h}_t, b_t = \mathbf{b}^\mathsf{T}\mathbf{h}_t$$

When using cosine similarity we train the network using a rank objective [13] that enforces matching by encouraging matching image and text to be similar to another up to a threshold λ. Let I^+ be a match to h_T and I^- a mismatching image. Then:

$$\mathcal{L}(\mathbf{h}_t, \mathbf{h}_i^+, \mathbf{h}_i^-) = \max(0, \lambda + s(\mathbf{h}_t, \mathbf{h}_i^-) - s(\mathbf{h}_t, \mathbf{h}_i^+)))$$

When using model-based similarity we train the network using standard binary cross-entropy loss:

$$\mathcal{L}(\mathbf{h}_t, \mathbf{h}_i, y) = y \log s(\mathbf{h}_t, \mathbf{h}_i) + (1 - y) \log (1 - s(\mathbf{h}_t, \mathbf{h}_i))$$

We conduct a comparative evaluation to quantify the performance of both similarity approaches. Results are given in Table 2.3 for a setup with Flair word embeddings and Fashion DNA image embeddings.

Similarity function	median rank	recall@1	recall@5	recall@10	recall@20
Cosine	4	0.258	0.571	0.717	0.828
Model-based	**3**	**0.33**	**0.661**	**0.782**	**0.87**

Table 2.3: Impact of different similarity functions on retrieval performance.

We find that the model-based approach significantly outperforms our earlier cosine distance-based approach.

2.4 Final Architecture

Based on these findings, we choose two configurations for the final architecture:

Best Proprietary The first is a configuration that represents the best text-image retrieval model we developed. It uses model-based similarity in combination with “Flair-Fashion” embeddings to represent text and “Fashion DNA” to represent images. However, both these embedding models are proprietary to Zalando, so the model cannot be distributed among FashionBrain parters.

Best Non-Proprietary The second configuration represents the best text-image retrieval model that does not employ proprietary models or data. It also uses model-based similarity, but uses default “Flair” embeddings to represent text and a task-trained TransformerConvNet for images. This model can be distributed not only to FashionBrain parters but also to third parties.

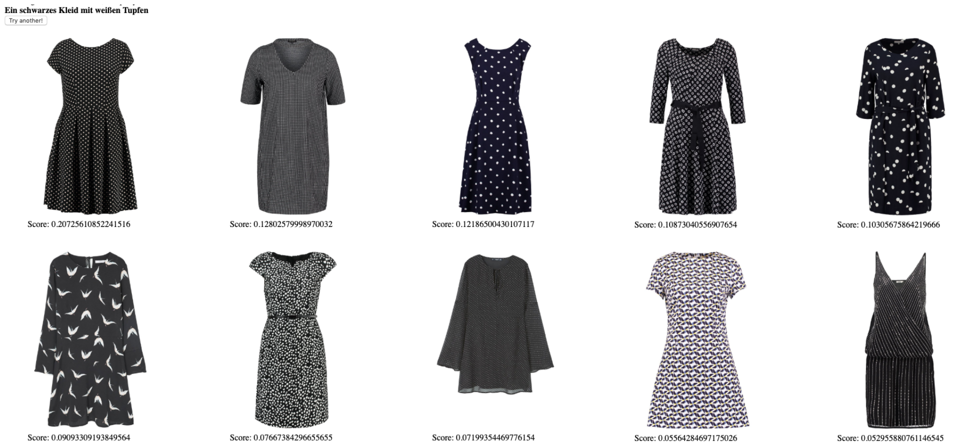
Both configurations significantly improve on the early earlier prototype presented in D6.3 which used only character embeddings for text, a ResNet-50 for images and cosine similarity. Our new model employs greatly improved text and image models and a model-based similarity function.

3 Demonstration

We present the final system as a text-image search demo that showcases how complex textual queries can be used to retrieve matching product images. Since we trained the final system using the publicly available FEIDEGGER dataset, queries need to be in the fashion domain of dresses. In this section, we first present screenshots illustrating the breadth of semantics covered by our approach. We then discuss how we disseminate our findings and the technology to third parties to enable them to reproduce our experiments or leverage this technology in their use cases.

3.1 Example Queries

Since basic categorical and attribute queries (e.g. color or brand) were already working in the earlier prototype presented in D6.3, this deliverable focuses on more complex queries. We do this to illustrate the improved functionality of the current system and to give an idea of its semantic coverage. We also discuss queries that are not yet well handled by the system to illustrate limitations that may be overcome with future research in natural language processing and computer vision.



Ein schwarzes Kleid mit weißen Tupfen

Score: 0.20725610852241516

Score: 0.12802579998970032

Score: 0.12186500430107117

Score: 0.10873040556907654

Score: 0.10305675864219666

Score: 0.09093309193849564

Score: 0.07667384296655655

Score: 0.07199354469776154

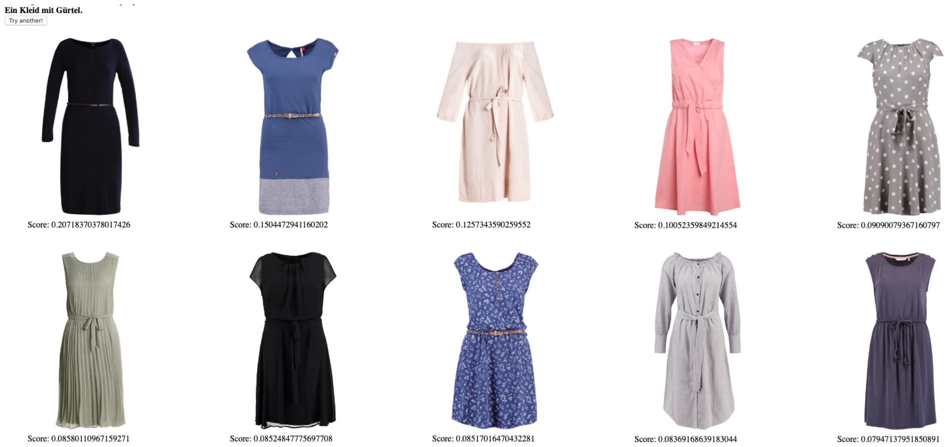
Score: 0.05564284697175026

Score: 0.05295580761146545

Figure 3.1: Search results for "*schwarzes Kleid mit weißen Tupfen*", which is "black dress with white dots" in English.

Multi-color query (see Figure 3.1). We begin with the query "*schwarzes Kleid mit weißen Tupfen*", which in English is "black dress with white dots". The query is complex in the sense that it mentions two colors (black and white), but each color pertains to a different aspect of the desired fashion item: The base color should be black, while a pattern of white dots should be visible. Thus, the text model has to decompose which color belongs to which attribute. On the image side, the core challenge is that "white dots" as a general concept may match a very diverse set of images. Dots may be very large or very small, numerous or sparse.

As Figure 3.1 shows, the system is able to identify many matching dresses that are black and are covered with white dots in various sizes or shapes. In particular, the top 5 matching queries (top row) are diverse and correct search results. We also note that some search results are partially wrong. Result 6 (bottom row leftmost) for instance is covered in stylized white birds instead of dots. The last three search results (bottom row right three) all diverge from the query to a certain extent. This indicates that the image model sees similarities between dots and reflections or stylized birds.

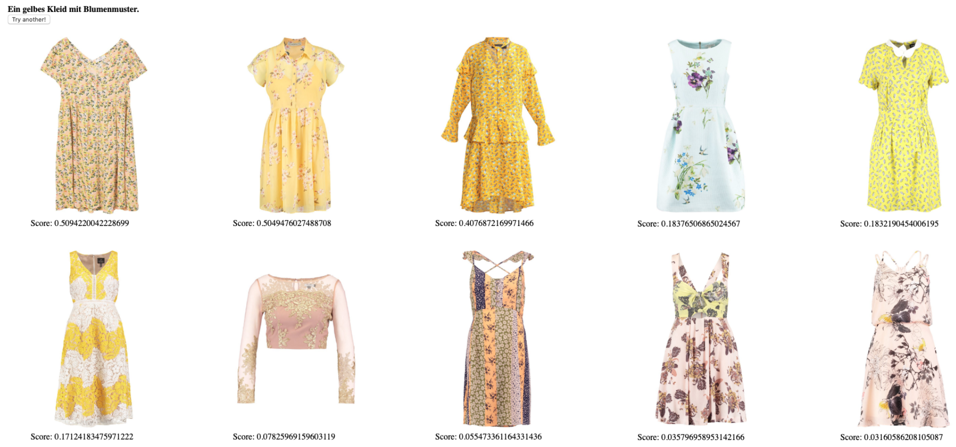

Ein Kleid mit Gürtel.

Score: 0.20718370378017426
Score: 0.1504472941160202
Score: 0.1257343590259552
Score: 0.10052359849214554
Score: 0.09090079367160797
Score: 0.08580110967159271
Score: 0.08524847775697708
Score: 0.08517016470432281
Score: 0.08369168639183044
Score: 0.07947137951850891

Figure 3.2: Search results for "*Kleid mit Gürtel*", which is "dress with a belt" in English.

Faint image features (see Figure 3.2). We now showcase the query "*Kleid mit Gürtel*", which in English is "dress with a belt". The difficulty of this query lies in the fact that certain features such as belts are both diverse (there are many different types of dress belt) and often do not stand out (belt dress color is often - but not always - the same as the dress itself). This query also leaves the color underspecified, meaning that we now desire a diverse search results of any dress of any color or type that has a belt. As Figure 3.2 shows, the query is well handled. All ten search results have belts in various colors or shapes.

Rare feature combinations (see Figure 3.3). We also issue the query "*gelbes Kleid mit Blumenmuster*", which in English is "yellow dress with a flower pattern". The query is difficult for two reasons. First, flower patterns can take many different sizes and shapes and so are a highly diverse image feature. Second, and more importantly, in Feidegger only a small number of dresses match this combination of features (yellow color and flower pattern), meaning that there are only few correct



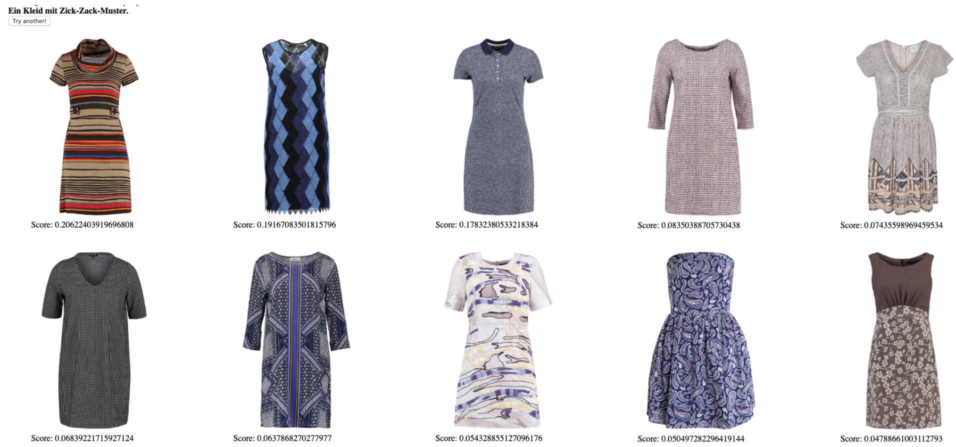
Ein gelbes Kleid mit Blumenmuster.

Score: 0.5094220042228699
Score: 0.5049476027488708
Score: 0.4076872169971466
Score: 0.18376506865024567
Score: 0.1832190454006195
Score: 0.17124183475971222
Score: 0.07825969159603119
Score: 0.055473361164331436
Score: 0.035796958953142166
Score: 0.03160586208105087

Figure 3.3: Search results for "*gelbes Kleid mit Blumenmuster*", which is "yellow dress with a flower pattern" in English.

search results to this query. Furthermore, this also means that there are few training examples for this composition of image features.

As Figure 3.3 shows, the query is mostly well handled. Four of the top five search results (top row) are indeed yellow dresses with flower patterns. The bottom row consists of many dresses that have flower patterns, but the lowest four matching search results (bottom row rightmost four) are not yellow. Our approach returns a score for each image that indicates how well it matches the query. We note that the scores for the four non-yellow dresses are very low. Depending on the use case, this enables us to filter these results and present only high-confidence search results to the customer.



Ein Kleid mit Zick-Zack-Muster.

Score: 0.20622403919696808
Score: 0.19167083501815796
Score: 0.17832380533218384
Score: 0.08350388705730438
Score: 0.07435598969459534
Score: 0.06839221715927124
Score: 0.0637868270277977
Score: 0.054328855127096176
Score: 0.050497282296419144
Score: 0.04788661003112793

Figure 3.4: Search results for "*Kleid mit Zickzackmuster*", which is "dress with a zigzag pattern" in English.

Rare semantics (see Figure 3.4). We now present an example query that does not yet work well, namely "*Kleid mit Zickzackmuster*", which in English is "dress with a zigzag pattern". As Figure 3.4 shows, only three returned dresses have a zigzag pattern, while the other dresses contain stripes or waveform patterns of different kinds. Like in previous queries, the concept of "zigzag" may be realized in many different ways, be it to separate hem from dress (top row rightmost image) or as a zigzag stripe pattern (top row second from left). However, the core difficulty here is that the term "zigzag" is only rarely mentioned in the training data, giving our algorithm little evidence with which to learn this concept.

3.2 Discussion

Our quantitative and qualitative evaluations indicate a strong general performance of our approach. In particular, we showed that text and image understanding modules are capable of producing meaningful representations for many concepts. However, our qualitative inspection of difficult queries also showed that there is a strong connection between system performance and available training data: Concepts that appear sufficiently frequently in the training data are generally well modeled, while the system struggles with rare concepts. These "long tail" semantics (i.e. semantic concepts that are rarely mentioned) are very challenging and are a focus of current research by FashionBrain partners and the broader research community.

Potential solutions involve either generating more training data (for instance with crowdsourcing process jointly developed by the FashionBrain consortium) or by researching "zero-shot" learning approaches that are capable of learning from small amounts of available data.

3.3 Dissemination

We publicly release many aspects of the system presented here to the academic community and for industrial use: The Feidegger [10] dataset is freely available online so that other groups can reproduce our experiments. The Flair [2] project is an open source text understanding library that has found significant uptake in both academic and industrial communities and has become one of the premier deep learning frameworks for NLP worldwide. By sharing this technology, we hope to further the state of research in text-image retrieval and enable scientific comparison of our approach to other approaches.

4 Conclusion

We presented the final architecture of our neural information retrieval system designed to match textual queries to images. Our system is fully operational for arbitrarily complex queries provided it is trained with suitable amounts of training data. The demo has been published on the FashionBrain web site, available at https://fashionbrain-project.eu/demo-on-textual-image-search/.

Bibliography

[1] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649, 2018.

[2] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. Flair: An easy-to-use framework for state-of-the-art nlp. In *NAACL, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics: System Demonstrations*, 2019.

[3] Alan Akbik, Tanja Bergmann, and Roland Vollgraf. Multilingual sequence labeling with one model. In *NLDL 2019, Northern Lights Deep Learning Workshop*, 2019.

[4] Christian Bracher, Sebastian Heinz, and Roland Vollgraf. Fashion dna: merging content and sales data for recommendation and article mapping. *arXiv preprint arXiv:1609.02489*, 2016.

[5] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[6] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[9] Benjamin Heinzerling and Michael Strube. BPEmb: Tokenization-free Pre-trained Subword Embeddings in 275 Languages. In Nicoletta Calzolari (Conference chair), Khalid Choukri, Christopher Cieri, Thierry Declerck, Sara Goggi, Koiti Hasida, Hitoshi Isahara, Bente Maegaard, Joseph Mariani, Hélène Mazo, Asuncion Moreno, Jan Odijk, Stelios Piperidis, and Takenobu Tokunaga, editors, *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki,

Japan, May 7-12, 2018 2018. European Language Resources Association (ELRA). ISBN 979-10-95546-00-9.

[10] Leonidas Lefakis, Alan Akbik, and Roland Vollgraf. FEIDEGGER: A multimodal corpus of fashion images and descriptions in German. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC-2018)*, Miyazaki, Japan, May 2018. European Languages Resources Association (ELRA).

[11] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[12] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543, 2014.

[13] Andrew Trotman. Learning to rank. *Information Retrieval*, 8(3):359–381, 2005.