APPLICATION OF MACHINE LEARNING-BASED NAMED ENTITY RECOGNITION FOR THE IDENTIFICATION OF NATURAL PERSON NAMES IN CORPORATE'S DOCUMENTS

Tom Magerman - MSc Business and Information Systems Engineering, PhD

Thijs Lemmens - Chief Product Officer @Xenit

Rawia Benhmida - ECM Engineer @Xenit

xenit
return on content

INTRODUCTION

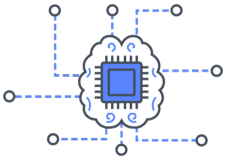
In this pilot study, we want to check the effectiveness of applying automated named entity recognition on a set of miscellaneous documents in a corporate document management system. The idea is to use machine learning (artificial intelligence) techniques to identify and extract natural person names from documents. This results in a list of person names and the position of where those person names appear in the document, and that information can be added to the document as meta-data.

We are interested in person name extraction as this is instrumental for the automated classification of GDPR-sensitivity of documents.

Indeed, the presence of natural person names is an important factor to derive GDPR-sensitivity, and having such meta-data available helps developing tools for automated GDPR-classification of documents in a corporate document management system.

Having those names available as meta-data is also instrumental in improving the quality of corporate search and retrieval systems; the presence of such meta-data can be used to automatically structure information and reveal links between documents. E.g. in an insurance context, this kind of metadata allows to centralise information around customers (all contracts of a person; all claims a person is involved in) and derive patterns and relationships (e.g. the same person popping up in multiple claims related to a limited set of contracts).

OBJECTIVES OF THE PILOT



The objectives are twofold. On the one hand, we want to find out whether machine learning-based named entity recognition methods and tools can indeed correctly identify private person names out of the box (using machine learning models that are readily available of the shelf, i.e. models trained on a large set of general documents). To do so, we compare the pre-trained model for English of **OpenNLP** (en-ner-person.bin) and **Stanford CoreNLP** (english.all.3class.caseless.distsim.crf.ser.gz).

On the other hand, we want to find out to what extent custom training, i.e. training on a particular document set, can improve the results and how training efforts (the number of cases available for training) influence the effectiveness. To do so, we compile training sets with an increasing number of documents from a custom corporate document set, derive classification models using OpenNLP and Stanford CoreNLP based on those training sets, and compare results.
For all models, both effectiveness (how many person names are correctly identified) as efficiency (how much calculation time does it take to get results) is compared.

To be able to evaluate and compare the effectiveness of machine learning-based methods, one needs to have a set of labeled documents, i.e. a set of annotated documents where every natural person name is labeled, to compare the results of the machine learning method with the real presence of natural person names ("ground truth").

2,616

DOCUMENTS

23,462

NATURAL PERSON NAMES

47,351

TOKENS

PROJECT SETUP

The starting point for our pilot is a corporate English document set where 2,616 documents were labeled by hand resulting in **23,462 natural person names** (as a natural person name can consist of multiple words or tokens – first name, middle name, last name - **47,351 tokens were identified** as being part of a natural person's name).

This set of annotated documents is used for training the named entity recognition model. As we want to see the impact of training efforts (the number of training cases used for training), multiple training sets with increasing size are compiled(n=100, 250, 500, 1000, 1500 and 2000 documents). To get results that are representative for the whole document set and not biased by the accidental selection of training cases, two independent – non-overlapping - training sets were compiled for every size. I.e. for every training size (n=100, 250, ...), n documents are randomly selected for one training set, and from the remaining documents again n documents are randomly selected for another training set.

As only 2,616 annotated documents are available, this independent selection is not possible for the training set with 1,500 and 2,000 documents (that would require a labeled dataset with 3,000 or 4,000 documents respectively). For the training set with 2,000 documents, only one training set was compiled, and for the training set with 1,500 documents, two training sets were compiled with as little overlap as possible. This results in 11 training sets.

PROJECT SETUP

The danger of (supervised) machine learning is that the system can only learn from the patterns that are present within the particular labeled training set. If the content of that training set (a random sample of documents) is not representative for the more global document set, the system will not grasp the latent patterns in the global document set. This will result in an optimal classification within the training set, but a bad classification of new documents outside the training set. This is called 'overfitting', i.e. the model is trained to perfectly grasp the patterns within the training set, but is not able to grasp patterns for documents outside the training set.

To test whether the trained models are general enough to be effective outside the training set, a test set is compiled with annotated documents that are not part of any training set, so the effectiveness of the models can be assessed for those 'new' documents (i.e. documents not used for the training).

As we are limited in the number of annotated documents (2,616), and many annotated documents are needed to assess the effectiveness of the training size (up to 2000 documents, preferable with two independent training sets per size level), only a small number of annotated documents is left to be used for testing. We decided to use two independent test sets to compensate for the low number of annotated documents left for testing.

This allow also for small test samples to assess the ability of the trained models to be applied on more general document sets (if the result of the two independent test sets are different, then we have the problem that the test results are biased because of the random selection of the test documents – and application on any new document set could again yield different results; if the results of the two independent test sets are similar,then we can conclude that the results are general enough to be applied on any other new document set). So, we compiled two independent test sets, each with200 randomly selected annotated documents that were not used for training.

PROJECT SETUP

To summarize: **11 training sets with an increasing number of documents** (n=100, 250, 500, 1000, 1500 and 2000 documents) were compiled - two independent sets for every size level except for the 2000 documents - from the 2,616 available labeled documents (documents where natural person names were manually identified).

From the remaining labeled documents, 2 independent test sets where compiled. The 11 training sets were used to train a model for both OpenNLP and Stanford CoreNLP, resulting in 22 models for the identification of natural person names. For each of those 22 models, the two test sets were used to assess the effectiveness of those models. I.e. for each model, precision (how many of the natural person names identified by the model are indeed natural person names) and recall (how many natural person names are not identified by the model) is calculated for the two test sets, resulting in 44 precision and recall measurements.

THE RESULTS

Training time for OpenNLP and CoreNLP models

	Training Set 1		Training Set 2	
Model	OpenNLP	CoreNLP	OpenNLP	CoreNLP
100	00:01:03	00:02:26	00:00:42	00:01:56
250	00:01:58	00:05:28	00:02:20	00:06:57
500	00:05:01	00:10:46	00:03:26	00:12:14
1000	00:08:16	00:16:45	00:06:21	00:17:26
1500	00:11:18	00:20:30	00:10:41	00:19:08
2000	00:15:24	00:39:31	(*)	(*)

(*) Only one training set available with 2000 documetns

Extraction time OpenNLP and CoreNLP models

Test Set 1		Test Set 2	
OpenNLP	CoreNLP	OpenNLP	CoreNLP
20s	**58m**	**19s**	**1h20m**

THE RESULTS

Precision (P) and Recall (R) of OpenNLP and CoreNLP for pre-trained model and custom trained models

Model	Test Set 1				Test Set 2			
	OpenNLP		CoreNLP		OpenNLP		CoreNLP	
	P	R	P	R	P	R	P	R
Pre-trained	0.30	0.57	0.70	0.64	0.25	0.50	0.71	0.61
100-1	0.84	0.43	0.93	0.46	0.95	0.53	0.97	0.51
100-2	0.88	0.40	0.93	0.36	0.93	0.43	0.96	0.41
250-1	0.95	0.49	0.94	0.54	0.97	0.56	0.95	0.59
250-2	0.94	0.53	0.91	0.59	0.95	0.50	0.93	0.60
500-1	0.93	0.59	0.93	0.62	0.96	0.63	0.95	0.65
500-2	0.82	0.67	0.94	0.67	0.88	0.71	0.93	0.67
1000-1	0.93	0.61	0.93	0.72	0.97	0.64	0.96	0.80
1000-2	0.91	0.67	0.93	0.78	0.96	0.72	0.96	0.81
1500-1	0.92	0.63	0.92	0.76	0.97	0.69	0.96	0.80
1500-2	0.91	0.71	0.93	0.81	0.96	0.77	0.96	0.81
2000	0.70	0.83	0.94	0.83	0.78	0.86	0.96	0.84

CONCLUSIONS ON EFFICIENCY

Training and extraction time

OpenNLP clearly outperforms CoreNLP with respect to training and extraction time. Depending on the training set, OpenNLP is two to three times faster than CoreNLP for training. Training times are slightly sublinear with the number of documents used for training for both OpenNLP and CoreNLP.
For the extraction, differences are huge. It takes about 20 seconds to process 200 documents for OpenNLP, and at least one hour for CoreNLP. Extraction times are not influenced by the model used (model trained on 100, 250, …, documents). All calculation times are based on a single core.

CONCLUSIONS ON EFFECTIVITY

Precision and Recall

For the pre-trained models, we see significantly better results for CoreNLP over OpenNLP. Striking are the very low precision results for OpenNLP (more than 70% of person names identified by OpenNLP are not personal names), making the pre-trained CoreNLP model useless for practical applications. Although CoreNLP results are better (70 to 71% precision and 61 to 64% recall), results are also not good enough for fully automated systems.Training a custom set of documents greatly improves results with respect to precision and recall.

CONCLUSIONS ON EFFECTIVITY

Precision and Recall

For OpenNLP results raise above 90% precision and above 80% recall (striking, however, is the drop in precision for the training set with 2000 documents). But again CoreNLP yields better results; while precision results are more or less inline, recall results are significantly better for almost every custom training set, with differences up to 10 percentage points.

However, for the biggest training set (2000), recall results of OpenNLP are slightly better. Again, the striking point is the drop in precision rates for this training set for OpenNLP (about 20 percentage point drop compared to the training sets with 1500 documents). This sudden drop might be a coincidence, due to the random selection of documents for this training set. If that is the case, then it might be that OpenNLP and CoreNLP perform similarly for the bigger training set.
As for the impact of the training size, we observe a steady increase in recall when more documents are used for training. Striking is the sharp increase in precision for even the smallest training size, but the fluctuating pattern for increasing training size. As for the 'stability' of the results (would different samples for training and testing yield different results, i.e. to what extent can these results be generalized) we observe that results for test set 2 are better compared to the results of the test set 1, but that pattern of the results is similar. For the variation in the selection of test sets we see very similar results for precision, but a significant difference for recall.

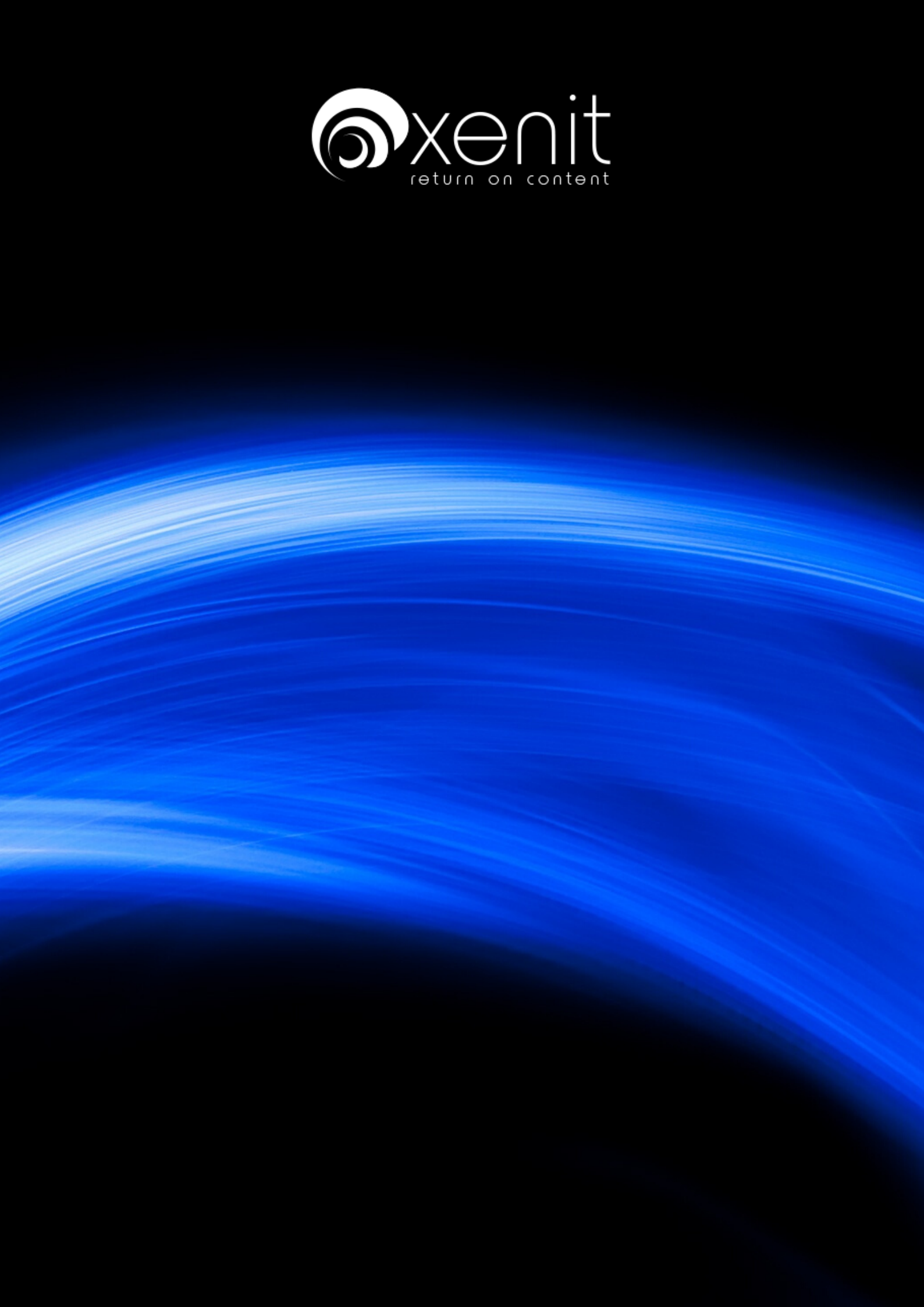
CONCLUSIONS ON EFFECTIVITY

Precision and Recall

Overall we tend to conclude that CoreNLP performs better. For the biggest training set, recall is similar, but precision not because of the striking drop in precision for OpenNLP. This might be due to coincidence – the random selection of documents for this training set – and if that is the case it might be that results for OpenNLP and CoreNLP tend to converge for larger training sets.

The default pre-trained models are not good enough, but even small custom training sets yield significantly better results. However, recall rates remain problematic. Low precision rates can be improved by presenting samples to the end-user in a production system, but low recall rates are difficult to remedy in a practical setup (you cannot ask for confirmation for something that you did not find). However, we do observe gradually increasing recall rates for increasing training size (as opposed to a fluctuation pattern for precision rates), which raises the question if recall rates cannot be pushed above 95% by increasing the training size to a few thousand documents (which is still feasible to manage).

The difference in results between the two test sets and two training sets (two training sets for every training size) also imply that larger training sets might be appropriate.


xenit
return on content