FFI Norwegian Defence Research Establishment

21/00022

FFI-RAPPORT

Content search in large text corpuses using natural language processing

Bernt Ivar Utstøl Nødland
Hallvar Gisnås
Henrik Gråtrud
Vidar Benjamin Skretting

Content search in large text corpuses using natural language processing

Bernt Ivar Utstøl Nødland
Hallvar Gisnås
Henrik Gråtrud
Vidar Benjamin Skretting

Norwegian Defence Research Establishment (FFI) 28 April 2021

Keywords
Dyp læring
Jihad
Maskinlæring
Terrorisme

FFI report
21/00022

Project number
1537

ISBN
978-82-464-3376-9

Approvers
Ingebjørg Kåsen, *Research Manager*
Trygve Sparr, *Research Director*

The document is electronically approved and therefore has no handwritten signature.

Summary

Analysts and researchers are facing an ever increasing amount of information. Finding ways to identify relevant information on fuzzy topics and concepts can thus accelerate the analyst. We investigate the method of using deep learning for semantic content search in a large text corpus. We test several state of the art models, such as ULMFiT and transformer based models. Deep learning models leverage large public corpuses to achieve a comprehensive understanding of language, such as next word prediction, to aid it's prediction of relevance. We compare them to a baseline of keyword search on a test case of approximately 50 000 articles from *Jordan Times*, where we try to identify articles about jihadist terror plots. We find that the best deep learning models outperform keyword search, indicating that these techniques could provide a useful tool for the analyst. However, they require effort to set up properly, and are much more complex compared to the baseline. We recommend to do further testing of these methods, both in English and in other languages.

Sammendrag

Analytikere og forskere står overfor en stadig økende mengde informasjon. Derfor kan det å finne nye måter å identifisere informasjon om spesifikke emner og konsepter akselerere analytikeren. Vi undersøker teknikker fra dyp læring for å søke etter spesifikt semantisk innhold i en stor tekstsamling. Vi tester flere av de nyere tekstforståelsesmodellene, som ULMFiT og transformer-baserte modeller. Dyp læring modeller bruker store offentlige tekstkorpus for å oppnå grundig forståelse av språk. Vi sammenligner dem med stikkordssøk på et testtilfelle bestående av ca. 50 000 artikler fra *Jordan Times*, der vi prøver å finne artikler om jihadistiske terrorplot. Vi finner at de beste modellene basert på dyp læring gjør det bedre enn stikkordssøk. Dette indikerer at disse teknikkene kan være nyttige for analytikere. Et forbehold er at disse teknikkene krever en del innsats for å sette opp og er mye mer komplekse enn stikkordssøk. Vi anbefaler å gjøre mer testing av disse metodene, både på engelsk og andre språk.

Contents

(U) Summary 3

(U) Sammendrag 4

1 Introduction 7

2 Case study: Jordan Times 8
2.1 Dataset 8

3 Methods 10
3.1 Baseline: Keyword search 10
3.2 Topic Modeling 10
3.3 Deep learning for natural language understanding 10
3.3.1 ULMFiT 11
3.3.2 Sentence similarity 12
3.3.3 Transformer classifiers 12

4 Implementation details 13
4.1 Topic Modeling 13
4.2 ULMFiT 13
4.3 Sentence similarity 14
4.4 Transformer classifiers 14

5 Results 16
5.1 Assessment of ranked lists 16
5.2 Evaluation of methods 17

6 Ideas for future work 19
6.1 Possibility of refining the methods 19
6.2 Possibilities of combining methods 19
6.3 Self-supervised learning 19
6.4 Possibilities in other languages 19

7 Conclusion 21

References 22

Appendix

A Words used in keywords search 25

B Examples of articles 26

B.1 Reference articles used for Sentence similarity 26
B.2 Examples of relevant articles found 27
B.3 Examples of semi-relevant articles 28
B.4 Examples of irrelevant articles 29

1 Introduction

In many sciences (particularly in the humanities and social sciences), one frequently has to read through large bodies of text, looking for some specific type of content. Simple word searches are often not sufficient to find all instances of the content of interest, due to the large number of synonyms and different ways of writing the same semantic content. In this report, we investigate whether one can use recently developed techniques in natural language processing (NLP), a subfield of artificial intelligence(AI), to search a large corpus for documents containing certain semantic content.

Here we employ these techniques on a specific case: jihadist terror plots (i.e., attack plans) in Jordan. We selected this case because the TERRA project at FFI, which conducts research on terrorism and asymmetric threats, has an ongoing project focusing on such plots in Jordan (see, e.g., [NG19], [Nes19]). We gather approximately 50 000 articles from the newspaper *Jordan Times* and attempt to find the articles about jihadist terrorist plots.

We try several different NLP techniques and use them to rank the articles from Jordan Times according to relevancy for jihadist terrorist plots. We compare these methods to each other, and to a naive baseline based on a simple word search. The methods are evaluated quantitatively. We discuss possible extensions and refinements in the methods, as well as the practical challenges in applying these methods on the specific case.

The study in this report is done for articles in English only. The reason is that the models and techniques from NLP are most developed in English, thus they are easiest to use for a first case study. However, we would also be interested in employing similar NLP techniques in other languages for future applications. In particular, for the case of Jordan, it would be very useful to have working models in Arabic, since most Jordanian newspapers write in Arabic. From an AI perspective, it would also be interesting to see the difference in performance between English and a language with less available data. We discuss possibilities and challenges in trying to apply these methods to such a language (e.g., Norwegian or Arabic).

2 Case study: Jordan Times

Nesser and Gråtrud [NG19] have written a comprehensive study on jihadist plots in Jordan. Here plots are defined as plans to carry out terrorist attacks. They include foiled, failed, and launched plans. We employ jihadist and jihadism to describe transnational Sunni militants associated with groups such as al-Qaida and the Islamic State (IS).[1] A significant part of the work in performing such a study consists of reading through thousands of newspaper articles to compile a list of all terrorist plots within Jordan. We thus have a case where, if we could successfully apply NLP techniques to analyze data, one would expect a significant increase in efficiency for the analyst.

2.1 Dataset

We scraped the archive of *Jordan Times*. There are online articles in the archive from 2014 up to 2019. Using the python package BeautifulSoup,[2] which is helpful for reading html and css code, we iterate through the archive by date. We save all articles which are either local or regional (meaning that they are about Jordan or the neighbouring area, respectively). We record the title, subtitle (if available), text, date and category (local/regional). In total, we find 52926 articles, where 37033 are local and 15893 are regional.

A remark on the dataset is here in order: One might think that the dataset comprises all articles by *Jordan Times* in the given time period. However, we know that some articles from the period did not appear in the archive: Beforehand, we had a collection of 41 relevant (to jihadist plots) articles from the time period. Of those, 38 are in our dataset. We do not know whether there is a systematic reason for which articles are excluded from the archive. However, for our purposes, this is not very important. The dataset consists of most local and regional articles in *Jordan Times* in the period 2014-2019.

The articles we want to identify are those about jihadist terrorist plots within Jordan. All articles about this topic should be categorized as «local», however the regional articles are useful to include in the dataset to compare the of the various methods performance on identifying relevant cases (will the algorithm rank articles about terrorism within Jordan higher than those about terrorism in, e.g., Syria?). This dataset is also interesting to study from the perspective of fuzzy labels: An article about a terrorist attack or plot should be considered semi-relevant if it is not clear from the article whether the motivation is jihadism or something else. Similarly, attacks outside Jordan should be considered less relevant than those inside Jordan. A big challenge in this case study is figuring out a precise and reasonable way to evaluate results, since we do not really know how many relevant articles exist.

Based on a preliminary inspection of the dataset, we estimate that less than 1 % of the articles are relevant, as defined above. We think that between 1 and 3% of the articles are semi-relevant, as defined above. The dataset as collected is not labeled. For some deep learning applications we need a labeled dataset, thus we manually labeled 585 articles as relevant or not relevant. The labeled dataset was chosen as follows: we labeled the 38 articles we already knew were relevant. Moreover,

[1] Jihadists wage a global armed struggle to topple regimes in the Muslim world. Their ultimate aim is the re-establishment of the caliphate and the application of Islamic law. For more on jihadism, see, e.g., Thomas Hegghammer, Jihad in Saudi Arabia: Violence and Pan-Islamism since 1979 (Cambridge: Cambridge University Press, 2010), 1-15; Anne Stenersen "Jihadism after the 'Caliphate': towards a new typology," British Journal of Middle Eastern Studies (2018): 1-20.

[2] https://www.crummy.com/software/BeautifulSoup/

while doing initial test runs using only articles from 2017 that were labeled as local, we analyzed preliminary output lists, yiedling 168 new articles which we labeled as relevant or irrelevant. We also chose 400 random articles to label as relevant or irrelevant. The articles which we considered as semi-relevant were, for the time being, left unlabeled. A consequence of this process is that in the labeled dataset there are disproportionally many articles from 2017, a fact that might have unintened consequences for the results. In the labeled subset there are 57 articles labeled as relevant, thus approximately 90% is labeled as not relevant and 10% as relevant. In the future it could be interesting to label also semi-relevant articles (say as 0.5 on a scale from 1=relevant to 0=irrelevant) and see whether this improves the methods.

3 Methods

3.1 Baseline: Keyword search

As a baseline, we consider the naive method of simply searching for keywords certain to be relevant for jihadist terrorist plots (examples are «terrorist», «state security court», «plot», «daesh», see Appendix A for the full list). There are several ways one could do this. We choose the following: we rank articles by the fraction of the words in an article which are among the selected relevant words. A different way would be to count the number of occurrences of the relevant words; however, this skewers the ranking in favour of lengthy articles. Since the length of the articles varies quite a lot, ranking by the largest fraction seems better.

Of course, this method is not giving an optimal result for how well keyword search could possibly work; if one spends a lot of time trying different things and getting to know the corpus, one would almost certainly be able to get better results. However, the analogous statement is also true for any other method we apply: Given enough tuning of parameters, it could be made better. Thus we adopt the philosophy in this test study of not fine-tuning any of the methods particularly much, but rather use a first reasonable attempt as a comparison. After all, the more time one needs to spend fine-tuning and/or knowing domain-specific details of the dataset, the less time-saving the method will be for the user.

3.2 Topic Modeling

Topic modeling is an unsupervised clustering technique for documents. The goal is to separate some corpus of text into semantically meaningful collections of documents, also referred to as topics. This means that an accurate topic model could give insights into a large corpus which would otherwise take years to read in its entirety.

There are several proposed algorithms to perform topic modeling for text. We have used Latent Dirichlet Allocation (LDA) [BNE03] in this analysis. LDA is a generative probabilistic model, and assumes that documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words. The algorithm is unsupervised, but requires the number of topics as an input to the algorithm.

In the best case scenario topic modeling can be extremely useful and efficient. But topic modeling is also notoriously finicky. It requires careful text preprocessing techniques, to reduce complexity and remove noise. It also has several hyperparameters that requires tuning to achieve good results. The algorithm is also unstable. This means that shuffling input data or changing the random seed can alter the results significantly. [AFM18] describe some of these issues.

3.3 Deep learning for natural language understanding

In recent years, techniques from machine learning, and more specifically deep learning, have begun to dominate the field of natural language processing. In many practical tasks such as translation, text classification (used, among other things, by customer support chatbots) and text search, these techniques are now widely being used, either as a supporting mechanism or as the main program.

One of the most common challenges using deep learning is that many techniques require a large dataset that is labeled. In other words, someone has to first spend significant time manually labeling a dataset, which is subsequently used to train a neural network. To avoid this problem, much

of current research in deep learning for natural language processing is directed towards transfer learning, where one trains the model on a different task, where there is more available training data, which requires that the model "understands" similar structures that one requires to solve the original task. Thereafter, one employs this model also on the original problem, thereby circumventing the need for a (large) training dataset of its own.

In the field of NLP, there has recently been significant progress using transfer learning: One trains large language models on several different tasks like predicting the next word in a sentence, sentiment analysis, sentence similarity, and so on. In this process, the model is required to learn some of the structure in the language, and it will perform quite well even on some tasks and/or datasets that it has not been trained for. It is customary to afterwards fine-tune such a general model on the data of the target task.

There are essentially three competing types of neural network models used that process natural language: The traditional recurrent models like recurrent neural networks (RNN) [RHW86] and long short term memory (LSTM) [HS97], models based on convolutional neural networks [LeC+89] in one dimension, and the more recent transformer models that rely on self-attention [Vas+17], that enables the model to attend to the most relevant parts of the data. For a given task, it is not really clear what kind of network architecture is best. Most very large state of the art models are transformer models. The most well-known such models are BERT (text understanding) [Dev+18] and GPT-2 (text generation) [Rad+19].

Below we describe the various deep learning techniques we use to find semantic content in the corpus. Some of these are semi-supervised methods, meaning that, in addition to unsupervised pre-training on other data, they are trained on the labeled subset of the dataset. Others are purely unsupervised, never using our labeled dataset at all. To keep the output lists of relevant articles comparable, we thus test all of these methods on the unlabeled subset of the dataset (approximately 99 % of the dataset is unlabeled).

3.3.1 ULMFiT

ULMFiT (Universal Language Model Fine-tuning for Text Classification) [HR18] was among the first methods to demonstrate successful transfer learning for text. The method consists of two main steps: First, it trains a language model whose task is to predict the next word in a sentence on a large general purpose corpus, such as WikiText-103 [Mer+16]. The target corpus will most likely have a different language style than a typical Wikipedia article, so the method fine-tunes the language model on the target corpus. Note that training the language model from scratch on the target corpus requires that this target corpus is very large. In practice, this is often not the case, so using a pretrained language model is a necessary step to make this technique work. Secondly, it reuses the encoder from the language model, in other words the model without the the final prediction layer, to turn the text of a given document into a feature vector, and uses this as the input to a basic classifier to predict the category of a document. ULMFiT also recommends a set of training procedures such as discriminative fine tuning of the language model and gradual unfreezing for the classifier. In other words one initially trains only the last network layer while keeping the rest fixed and as training progresses one gradually trains more and more layers. This ensures robust training, and helps avoid catastrophic forgetting. ULMFiT demonstrated state-of-the art performance on several tasks upon release, but has since been superseded by other models, such as transformers. ULMFiT is a fairly small model though, and requires relatively little compute power compared to the recent transformer models.

3.3.2 Sentence similarity

One way to use deep learning to search for semantic content in a corpus is via using a model trained to detect sentence similarity. A sentence (or article) is mapped by a neural network to some vector. The idea is to train the network on data that, explicitly or implicitly, forces sentences with similar semantic content to map to close vectors (under some metric). Common such tasks could be classifying the sentiment of a sentence (as, say, positive or negative), or classifying pairs of labeled sentences as relevant or irrelevant to each other, or training to decrease the distances of similar sentences.

There are several different such pretrained models in the literature, available for download. We here try two such models: The Sentence Transformers (ST) [RG19] and Universal Sentence Encoder (USE) [Cer+18].

3.3.3 Transformer classifiers

Another technique to measure similarity is the following: Start with a transformer model which takes a sentence/article as input and outputs some feature vector, as above. Put a classification layer on top of that, which classifies articles as relevant or irrelevant. Now train the model on the labeled dataset to classify articles as relevant or not. In the test case, we make classifiers on top of the sentence similarity models ST and USE.

4 Implementation details

4.1 Topic Modeling

The first step for topic modeling is proper preprocessing. We used the spaCy library for lower casing all text, tokenisation, lemmatisation and removing spaCy's default stop words and punctuation.[3] We then did two iterations of inspecting the keywords from an LDA-model and added any unsuitable words to a list of custom stop words. Finally we also removed the least and most common words from the text.

Then, we estimated bigrams, that is pairs of tokens which it makes more sense to treat as one token, e.g. "New York". We estimated bigrams rather conservatively with the gensim library,[4] and inspected results to check that they mostly made sense.

Next, we grid searched topic models with the number of topics ranging from 10, 20, ..., 100. We used gensim's default hyperparameters, but set the number of passes to 50. We then used pyLDAvis[5] to inspect each topic model. We used a salience of 0.6 as suggested by [SS14] and manually inspected models until we found a topic that had keywords that matched the keywords outlined in, Section 3.1 and where central keywords such as 'terrorism' mostly was associated with that particular topic. We ended up selecting a topic model with 50 topics. The top 10 keywords from the chosen topic (with salience of 0.6) were: attack, group, terrorist, video, daesh, terror, terrorism, extremist, security, al_qaeda.[6] The final step is to rank all the articles from the corpus according to how much they are associated with this particular topic.

Note that this process has many steps that can be refined, and it also has a manual aspect in actually selecting a model that looks relevant. The resulting ranking of articles is highly dependent on the effort invested, and is likely not particularly reproducable. I.e. results will most likely vary with the analyst performing the analysis.

4.2 ULMFiT

The fastai v1 library[7] was used to implement ULMFiT, and the code is based on the code from one of the 2019 lectures of the fast.ai deep learning course.[8]

First, we process our target corpus with the spaCy tokenizer. The fastai library also adds a few special tokens.[9]

The language model we used was an AWD-LSTM [MKS18] pretrained on English WikiText-103. The language model was then fine-tuned on the target corpus for 10 epochs. Note that we fine-tune on the entire corpus, not only the small subset of labeled documents. Total training time using mixed precision training (floating point 16-bit) is around 30 minutes. It should be possible to reduce this further with a factor of around 2-3 using a QRNN architecture [Bra+19], but this has not been tested in this analysis.

[3] https://spacy.io/

[4] https://radimrehurek.com/gensim/

[5] https://github.com/bmabey/pyLDAvis

[6] Note the _ that comes from the process of generating bigrams, i.e. "al" and "qaeda" often appears together and is thus correctly recognized as a bigram.

[7] https://docs.fast.ai/

[8] https://github.com/fastai/course-v3/blob/master/nbs/dl1/lesson3-imdb.ipynb

[9] https://docs.fast.ai/text.transform.html#Tokenizer

Secondly, we create a classifier by reusing the encoder from the language model and adding two blocks, each consisting of a fully connected layer, followed by batch normalization [IS15], the ReLU activation function and dropout [Sri+14]. The classifier is first trained for one epoch in several steps. In each step we gradually unfreeze one layer of the classifier, starting with the final layer. Finally, we train for 10 epochs with all layers unfrozen. The classifier architecture is fairly small and only requires a few minutes of total training time.

Since the training set is quite imbalanced, we also inspect the confusion matrix to check if the classifier only predicts the majority class. In this particular case, the classifier performs well, so we use results as is.

4.3 Sentence similarity

We try two different sentence similarity neural networks: ST and USE. Both of these take as input a sentence and outputs some vector representation of the sentence. We then measure the distance between various sentences. We measure the distance in cosine similarity. In other words, the smaller the angle between the two vectors is, the more similar they are considered to be.

USE is trained from scratch on various natural language understanding tasks on the sentence level. ST starts from a pre-trained BERT model and fine-tunes on sentence level tasks. Both can take arbitrarily large texts as input. We thus input whole articles instead of sentences. We use the pre-trained implementation of ST found at the authors' github repo,[10] as well as the pre-trained USE found at Tensorflow Hub.[11] Both are implemented in TensorFlow.[12]

The strategy for our use case is the following: Start with some reference articles already identified as relevant to jihadist terrorist plots. For each of the other articles in the corpus, we measure the similarity between it and the reference articles. The average of these is the relevancy score. In our test case, we chose 31 reference articles. From each model we obtain a ranked list of the unlabeled articles, ranked by how similar they are to the 31 reference articles.

How well this method works is of course highly dependent on the 31 articles chosen: We do not really know whether they are representative of an average article on terrorist plots in the dataset. However, this is in some sense a feature and not a bug: In a realistic use case of these methods, the user should be able to find some articles that are deemed relevant and query the method for similar articles. However, it is not reasonable to expect the user to have a very precise preconceived notion of what are representative samples. It would be interesting to study to what degree the performance and rankings change based on a different set of reference articles.

4.4 Transformer classifiers

We put a binary classifier on top of either one of ST and USE and train this to classify articles as relevant or irrelevant based on the labeled subset of the articles. The final layer outputs a single number in the range [0,1], where 1 means relevant and 0 means not relevant. Then, we classify all articles with a relevancy score above a certain threshold (naively we put it as 0.5) as relevant, and the others as not relevant. After training this classifier, we input all unlabeled articles into the classifier and get an output list sorted by the relevancy score.

[10] https://github.com/UKPLab/sentence-transformers

[11] https://tfhub.dev/google/universal-sentence-encoder/

[12] https://www.tensorflow.org/

Since our labeled dataset is unbalanced (90% are labeled as irrelevant), the first attempt at training a classifier simply classifies everything as irrelevant, since it then achieves 90 % performance. To remedy this, we use class weighting; i.e., the relevant articles are weighted higher in the loss function. When we do this, the classifier starts classifying articles also as relevant. We train on the labeled dataset for 12 epochs. The number of epochs was chosen by the method of early stopping: we observe that after a certain number of epochs we overfit and stop training before that.

5 Results

For each of the studied methods, we output a list of ranked articles (among the unlabeled articles) assessed as most relevant for terrorism plots. We have no unique way of evaluating the performance of these lists, but we believe the following method gives a reasonable assessment: We choose a cutoff of the top 100 articles from each list. The experts on terrorism/jihadism get the list and label the examples from each list as relevant, irrelevant, or semi-relevant (including a reason for it being semi-relevant).

In total we identified 68 new relevant articles, 272 semi-relevant and 78 irrelevant articles from the seven top-100 lists. This means that there was signficant overlap among some of the lists. In Table 5.1 we see, for each category, how many different lists the articles in that category were in. We see that there is significant overlap among the relevant articles; more than half are in 2 or more lists, and approximately one third of the articles are in 5 or 6 lists.

	Relevant		**Semi-relevant**		**Irrelevant**	
Identified by 1 method	30	(44.1%)	191	(70.2 %)	76	(97.4 %)
Identified by 2 methods	6	(8.8 %)	42	(15.4 %)	2	(2.6 %)
Identified by 3 methods	3	(4.4 %)	18	(6.6 %)	0	(0 %)
Identified by 4 methods	6	(8.8 %)	15	(5.5 %)	0	(0 %)
Identified by 5 methods	14	(20.5 %)	4	(1.5 %)	0	(0 %)
Identified by 6 methods	9	(13.2 %)	2	(0.7 %)	0	(0 %)

Table 5.1 The table shows the overlap of identified articles for each method. E.g. approximately one third of the 68 relevant articles were identified by five or more methods. Irrelevant articles, on the other hand, have minimal overlap between methods. Note that no article was identified by all seven methods for any category.

5.1 Assessment of ranked lists

Table 5.2 lists the classification of the top 100 articles by method.

Method	**Relevant**	**Semi-relevant**	**Irrelevant**
Word search	29	41	30
Topic modeling	7	91	2
ULMFiT	39	60	1
ST	19	48	33
USE	41	57	2
ST classifier	28	60	12
USE classifier	36	64	0

Table 5.2 The number of relevant/irrelevant/semi-relevant articles in the top 100 articles by each method.

We see that the highest performing lists are those coming from ULMFiT, USE and USE classifier. These all do approximately equally well having 40% relevant and the rest semi-relevant. We do not know how many relevant articles are in the dataset, so we do know how many articles exist which

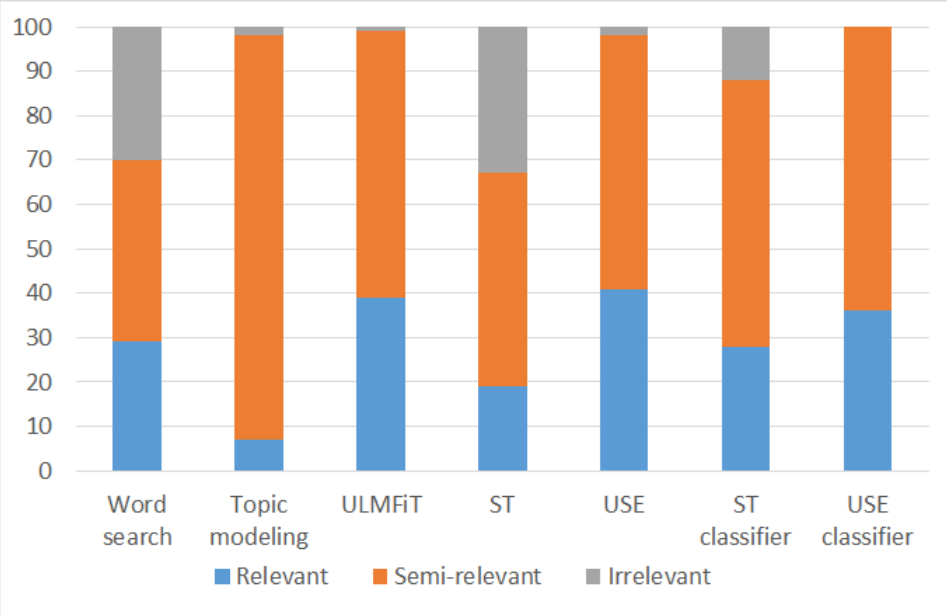


Figure 5.1 Distributions of categories of articles in top 100 lists, by method.

would be classified as relevant. These lists are regarded as high performing, since they contain almost exclusively articles which are to some degree interesting for someone studying terrorism in Jordan.

The lists given by word search and ST are approximately performing similarly at the bottom end of the scale, with ST being worst of all. They contain by far the highest number of irrelevant articles, implying that the methods are quite unstable in what they are looking for. Interestingly, the list provided by ST classifier is significantly better than that of ST, having drastically reduced the irrelevant articles on the list. The list given by word search distingusihes itself in one positive respect, observed by the analysts: It almost excusively contains articles about events inside of Jordan.

The outlier is the list provided by topic modelling: almost all articles are semi-relevant. These articles are often interesting for the analyst who is studying terrorism, but are rarely about terrorist plots inside of Jordan.

5.2 Evaluation of methods

Based on the above, we see that the baseline of naive word search is among the worst performers. This is encouraging, since it indicates that using more refined techniques are actually useful. In Appendix A we list the words used to search. A challenge in using keyword search is that we need to include diverse enough words to hope to be able to recall all the relevant articles, but this has the consequence of also finding a lot of completely irrelevant articles that include a subset of the words. However, this is also the cheapest method: no labeled data, modeling setup, or expertise is required.

The results for the method of topic modeling illustrates both its advantages and disadvantages: Almost all articles are relevant or semi-relevant, although a large majority are only semi-relevant. Thus we did manage to find a topic that was relevant for terrorism, which is the positive side, but it was not primarily about jihadist plots in Jordan. We think that this is related to the fact that there

are more articles on other specific topics concerning terrorism, e.g. terrorism in Syria, than there are on plots within Jordan. For a topic that does not have this property of being a minority among half-relevant cases, one would expect this method to work better. Of course, one could always spend time searching for a better topic that hits closer to the mark. However this is work intensive, and not guaranteed to be successful. Topic modeling is most likely better for exploration than for identification of specific content.

There is a large difference in the performance of ST and USE; USE is among the best performing models, while ST is worse than word search. This is somewhat surprising, because in our preliminary testing ST did almost as well as USE. However, we do observe that the classifier version of ST performs significantly better than word search, and only slightly worse than the best methods. ST classifier is simply ST with a binary classifier on top, trained on 500 examples for 12 epochs, but still it performs significantly better than ST. This illustrates one of the challenges in using deep learning, namely that it is hard to tell in advance how well a model will perform.

USE performed approximately the same with and without the classification layer. The good performance without the classification layer is actually somewhat surprising: The model has not at all been trained with the Jordan Times dataset, but it very accurately finds relevant similar articles to the reference articles. As the name indicates, this is the intention of the universal sentence encoder, but it is not obvious that it actually works, as in general one expects significant increase in performance for models which are fine-tuned on the specific dataset.

ULMFiT was also one of the best performing methods. This is not that suprising, given that it upon release beat state of the art models on many tasks, and moreover combines the techniques of transfer learning from large text datasets, fine-tuning on the specific dataset and adding a classification layer on top, utilizing the labeling as well. This is encouraging, since, due to it also coming with a training scheme, ULMFiT is relatively easy to use also in languages other than English. This assumes that one has a language model trained on a large text corpus (or trains one from the corpus) in the language in question.

The fact that word search was better than the deep learning methodologies in finding articles about terror plots inside of Jordan, illustrates a distincition in the methods: The deep learning models try to map similar sentences to similar vectors, thus terrorism in Jordan and terrorism in Syria will probably be regarded as similar. Word search, on the other hand, can apply Jordanian specific words, thus making a clearer distinction between these cases. Thus we hypothesise that word search might be better for some kind of queries (e.g., by being country specific), while deep learning models are better for others (e.g., in distiguishing terrorism from other kinds of criminal activity).

Both of the USE methods, together with ULMFiT, showed very good results in finding relevant articles in our text corpus. Thus we have two methods, both using deep learning, which outperform naive word search by a large margin. It is reasonable to think that the main conceptual reason for these techniques performing better than word search is that they are able to take the context in which a word appears into account when evaluating relevancy. This is impossible to do for a keyword search. A drawback to the deep learning methods is that they require more computational power and technical expertise to use.

6 Ideas for future work

6.1 Possibility of refining the methods

Each of the methods studied could be refined by spending time fine-tuning hyperparameters and methodology. Several techniques are frequently used to get slightly higher performance. One such technique is that of ensembling, which means training several different models and combining them in some way. Another possibility is making the classification model on top of the language models more sophisticated.

6.2 Possibilities of combining methods

Some of the methods we have tested could possibly be combined. For instance, ULMFiT trains a language model. It would be possible to fine-tune it to detect sentence similarity, seeing as ST for instance is a fine-tuning of the BERT language model. It would also be interesting to first use one of the methods to rank the articles according to relevance and then perform topic modelling on the top ranked articles, for instance the top 1000 or 5000, and see what performance topic modelling would give then. One would hope that if most articles are about topics related to terrorism, then topic modeling would more easily identify a topic about jihadist plots inside of Jordan. Another useful combination of methods could be to use keyword search to find likely relevant articles that we then label as relevant or irrelevant. Finding positives could be difficult when, as in our case, 99 % of all articles are irrelevant. Subsequently, we can train a classifier on the labeled dataset.

6.3 Self-supervised learning

A possible technique one could try for training classifiers, is that of self-supervised learning. This means that the model itself creates labels for the dataset, which it subsequently uses to train further. In our case we could easily do the following: Our classifier trained on the labeled subset outputs some relevancy score in the range $[0, 1]$ for every article. If we now consider this number as the label for that article, one could iterate and train on this new, larger, labeled dataset. This technique is reasonably new, but it shows promising results: It has been successfully applied to increase performance on image classification tasks [Xie+19].

6.4 Possibilities in other languages

An obvious limitation to the results in this study is that the deep learning techniques all rely on networks that are pre-trained on large text databases in English. For most other languages, there do not exist anywhere near the same amount of data, nor pre-trained language models.

There do however exist trained language models, such as BERT, in several languages other than English, even if they are not as good as the English model. There are also libraries that can be used to train such models from scratch. The most well-known library for doing this, as well as pre-trained language models in several languages, is huggingface's Transformers library.[13] This could be a good starting point for applying these techniques to other languages.

[13] https://github.com/huggingface/transformers

There is, however, some limitations: first of all, most language models are pre-trained on Wikipedia. The English Wikipedia is much larger than Wikipedia in most other languages, thus there is more data to pre-train on in English. For the sentence similarity methods, one also needs to have large datasets of labeled sentences to train the model to learn sentence representations. In most languages one does not have (large) enough such datasets.

One technique that could be explored is that of using multilingual language models. There are some promising results that some language models trained on several different languages could perform well for some tasks. Examples models include multilingual USE [Yan+19], LASER [AS18] and Multifit [Eis+19]. These models and techniques would probably be the best first attempt at doing something similar in languages other than English.

For the specific case of terrorism in Jordan one would naturally like to study articles in Arabic. Both multilingual USE and LASER are trained partly in Arabic, hence these would be a good first attempt. There are also some pre-trained models in Arabic available online, such as hULMona [ElJ+19], and, more recently, araBERT [ABH20]. Attempting to fine-tune these to our application using techniques from ULMFiT, or adding classification networks, seems like a sensible way to test these methods in Arabic.

7 Conclusion

In this study we have compared several different methods for seaching for specific semantic content in a large text corpus. The best results are achieved by deep learning models, in particular the models based on ULMFiT and the Universal Sentence Encoder. They perform significantly better than a naive keyword search, but come at the cost of requiring more computational resources and technical expertise. This indicates that these techniques from natural language processing could be useful to analysts trying to understand large quantities of text. An interesting possibility for future work would be to see what performance could be achieved for a language where there are smaller datasets and pre-trained models available, e.g. Arabic or Norwegian.

References

Agrawal, Amritanshu, Wei Fu and Tim Menzies (2018). 'What is wrong with topic modeling? And how to fix it using search-based software engineering'. In: *Information and Software Technology* 98, pp. 74–88. ISSN: 09505849. DOI: 10.1016/j.infsof.2018.02.005. arXiv: 1608.08176.

Antoun, Wissam, Fady Baly and Hazem Hajj (2020). *AraBERT: Transformer-based Model for Arabic Language Understanding*. arXiv: 2003.00104 [cs.CL].

Artetxe, Mikel and Holger Schwenk (2018). *Massively Multilingual Sentence Embeddings for Zero-Shot Cross-Lingual Transfer and Beyond*. arXiv: 1812.10464 [cs.CL].

Blei, David M, Andrew Y Ng and Jordan@cs Berkeley Edu (2003). *Latent Dirichlet Allocation Michael I. Jordan*. Tech. rep., pp. 993–1022.

Bradbury, James, Stephen Merity, Caiming Xiong and Richard Socher (2019). 'Quasi-recurrent neural networks'. In: *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR. DOI: 10.1117/3.633187.ch9. arXiv: 1611.01576.

Cer, Daniel, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Brian Strope and Ray Kurzweil (Nov. 2018). 'Universal Sentence Encoder for English'. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, pp. 169–174. DOI: 10.18653/v1/D18-2029. URL: https://www.aclweb.org/anthology/D18-2029.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee and Kristina Toutanova (2018). 'BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding'. In: *arXiv preprint arXiv:1810.04805*.

Eisenschlos, Julian, Sebastian Ruder, Piotr Czapla, Marcin Kadras, Sylvain Gugger and Jeremy Howard (Nov. 2019). 'MultiFiT: Efficient Multi-lingual Language Model Fine-tuning'. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, pp. 5702–5707. DOI: 10.18653/v1/D19-1572. URL: https://www.aclweb.org/anthology/D19-1572.

ElJundi, Obeida, Wissam Antoun, Nour El Droubi, Hazem Hajj, Wassim El-Hajj and Khaled Shaban (2019). 'hULMonA: The Universal Language Model in Arabic'. In: *Proceedings of the Fourth Arabic Natural Language Processing Workshop*, pp. 68–77.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). 'Long short-term memory'. In: *Neural computation* 9.8, pp. 1735–1780.

Howard, Jeremy and Sebastian Ruder (2018). 'Universal Language Model Fine-tuning for Text Classification'. In: *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)* 1, pp. 328–339. arXiv: 1801.06146. URL: http://arxiv.org/abs/1801.06146.

Ioffe, Sergey and Christian Szegedy (2015). 'Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift'. In: ed. by Francis Bach and David Blei. Vol. 37. Proceedings of Machine Learning Research. Lille, France: PMLR, pp. 448–456. URL: http://proceedings.mlr.press/v37/ioffe15.html.

LeCun, Y., B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard and L. D. Jackel (1989). 'Backpropagation Applied to Handwritten Zip Code Recognition'. In: *Neural Computation* 1, pp. 541–551.

Merity, Stephen, Nitish Shirish Keskar and Richard Socher (2018). 'Regularizing and optimizing LSTM language models'. In: *6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings*. International Conference on Learning Representations, ICLR. arXiv: 1708.02182.

Merity, Stephen, Caiming Xiong, James Bradbury and Richard Socher (2016). 'Pointer Sentinel Mixture Models'. In: *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*. arXiv: 1609.07843. URL: http://arxiv.org/abs/1609.07843.

Nesser, Petter (2019). *Foiled plots: the untapped data resource in terrorism studies*. URL: https://www.sv.uio.no/c-rex/english/news-and-events/right-now/2019/foiled-plots-the-untapped-data-resource.html.

Nesser, Petter and Henrik Gråtrud (Dec. 2019). 'When Conflicts Do Not Overspill: The Case of Jordan'. In: *Perspectives on Politics*, pp. 1–15. DOI: 10.1017/S153759271900389X.

Radford, Alec, Jeff Wu, Rewon Child, David Luan, Dario Amodei and Ilya Sutskever (2019). 'Language Models are Unsupervised Multitask Learners'. In:

Reimers, Nils and Iryna Gurevych (Nov. 2019). 'Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks'. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics. URL: http://arxiv.org/abs/1908.10084.

Rumelhart, David E., Geoffrey E. Hinton and Ronald J. Williams (1986). 'Learning Representations by Back-propagating Errors'. In: *Nature* 323.6088, pp. 533–536. DOI: 10.1038/323533a0. URL: http://www.nature.com/articles/323533a0.

Sievert, Carson and Kenneth E Shirley (2014). *LDAvis: A method for visualizing and interpreting topics*. Tech. rep., pp. 63–70.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever and Ruslan Salakhutdinov (2014). 'Dropout: A Simple Way to Prevent Neural Networks from Overfitting'. In: *Journal of Machine Learning Research* 15.56, pp. 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser and Illia Polosukhin (2017). 'Attention is All you Need'. In: *Advances in Neural Information Processing Systems 30*. Ed. by I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan and R. Garnett. Curran Associates, Inc., pp. 5998–6008. URL: http://papers.nips.cc/paper/7181-attention-is-all-you-need.pdf.

Xie, Qizhe, Minh-Thang Luong, Eduard Hovy and Quoc V. Le (2019). *Self-training with Noisy Student improves ImageNet classification*. arXiv: 1911.04252 [cs.LG].

Yang, Yinfei, Daniel Cer, Amin Ahmad, Mandy Guo, Jax Law, Noah Constant, Gustavo Hernandez Abrego, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope and Ray Kurzweil (2019). *Multilingual Universal Sentence Encoder for Semantic Retrieval*. arXiv: 1907.04307 [cs.CL].

A Words used in keywords search

Here we list the keywords used in the baseline method of keywords search.

- kalashnikov
- bomb
- terrorist
- terrorists
- terror
- plotting
- plotted
- stabbing
- cell
- state security court
- assassination
- gid
- sniper
- rocket
- armed
- foil
- thwart
- disrupt
- detect
- plot
- plan
- attack
- operation
- islamic state
- daesh
- al qaeda
- subversive
- firearms
- explosives
- weapons
- foil
- foiled
- state security
- court
- ssc
- sentence
- sentenced
- prison
- labour
- takfiri
- jihadist
- extremist

The list is a combination of words relevant for terrorist plots in general (terrorist, extremist, armed, attack), for jihadism (daesh, islamic state, jihadist) and for Jordan (state security court, ssc, gid (referring to the Jordanian General Intelligence Directorate)). The list illustrates the challenges of applying keyword search: since we are interested in articles satisfying several distinct criteria (bering about terrorist plot, jihadism and Jordan) we need to include words referring to all of these topics. However, this makes it likely that some of the articles we find are only related to one or two of these criteria, and thus sometimes are not relevant at all.

B Examples of articles

Here we list some examples of articles used or found in this study. We list the title in boldface, subtitle in italics (where applicable) and a couple of paragraphs. Sometimes this is the entire article, while often it is not, in which case we indicate this by ending the article with '...'.

B.1 Reference articles used for Sentence similarity

Based on prior work by Gråtrud we had 41 articles we knew to be relevant. These were used as reference articles for the sentence similarity models. A couple of examples are listed below.

'Terrorist' who stabbed British tourist condemned to 15-year-jail term. *SSC had previously handed him a life sentence but reduced it 'to give him second chance in life'*

AMMAN — The State Security Court (SSC) has sentenced a man to 15 years in prison after convicting him of stabbing a British tourist in Amman in February 2017.

The court had declared the defendant guilty of conducting terrorist acts by stabbing the British tourist in downtown Amman with a kitchen knife in February and handed him a life sentence.

However, the SSC tribunal decided to reduce the sentence to 15 years "because the defendant expressed regret and was young therefore deserved a second chance in life," a senior judicial source told The Jordan Times...

8 sentenced to 15 years each for planning 'Daesh-sponsored' terror attacks *3 separate plots to attack Marka airport, foreign embassies, soldiers, electricity lines with explosives and guns*

AMMAN — The State Security Court (SSC) on Wednesday sentenced eight men to 15 years in prison each in three separate terror-related cases, a senior judicial source said.

"The military court issued its ruling in three terrorism cases and handed all eight defendants, who are affiliated with or supporters of Daesh, the maximum punishment on charges of plotting to carry out subversive acts," a senior judicial source told The Jordan Times.

In one case, a man was convicted by the SSC of plotting to attack Marka Military Airport in Amman with an explosive belt, the judicial source said.

Baqaa GID office attacker sentenced to death

AMMAN — The State Security Court (SSC) on Thursday issued a death sentence against the gunman who killed five intelligence personnel in June in the terrorist attack on the Baqaa office of the General Intelligence Department (GID).

The Jordan News Agency, Petra, quoted Attorney General of SSC Brig. Gen. Ziad Adwan as saying the court found Mahmoud Masharfeh guilty of committing terrorist acts that led to the death of human beings, and committing terrorist acts using automatic weapons. The SSC issued a death sentence for each of the charges against Masharfeh.

The court also sentenced Sami Abu Omar to a one-year term, after amending the charge against him from the felony of selling weapons for illegal uses to the misdemeanour of selling weapons, Petra added.

B.2 Examples of relevant articles found

The top ranked article found by each method is shown below

Keyword search
Two sentenced to prison terms for promoting terrorist ideology

The State Security Court on Monday sentenced two defendants to two years in prison for promoting the ideology of terrorist groups in separate cases.

Topic modeling
Anti-terror campaign continues — gov't official

AMMAN — Jordan on Tuesday said it would continue security operations to arrest anyone who seeks to undermine the country's stability and security in the aftermath of Sunday's terrorist attack in Karak.

"We will defend our country, religion and people as well as the future of our children in the face of terrorist gangs and their heinous and inhumane acts that are prohibited by Islam," an official told The Jordan Times Tuesday.

"We all stand united behind our armed forces and security apparatuses to counter terrorism and crimes," the source added...

ULMFiT
Karak terror attack verdict to be issued within two weeks

AMMAN — The State Security Court (SSC) is expected to issue its verdict in the case of 11 defendants allegedly involved in the December 2016 terrorist attacks in Karak in the coming two weeks, a senior judicial source said.

The attack caused the death of 10 people, including four police officers and three gendarmes.

"The SSC is almost done with its deliberation and examination of the case and is expected to issue a verdict within the next two weeks," the senior judicial source told The Jordan Times, noting that the 11 defendants include 10 men who pleaded not guilty during their opening trial in previous months and one defendant who still remains at large.

Sentence Transformer
17 men found guilty of supporting terrorists

AMMAN — The State Security Court on Wednesday sentenced 17 defendants found guilty on 30 terrorist counts, including attempting to join terrorist groups like Daesh and Al Nusra Front, joining terrorist groups and promoting the terrorist ideologies.

The sentences handed down to the defendants ranged from three to 15 years imprisonment, the Jordan News Agency, Petra, reported.

The court also declared two defendants not guilty on charges of promoting the ideas of a terrorist group...

ST classifier and USE

Five Irbid cell terrorists sentenced to death 14 others receive varying jail terms

AMMAN — The State Security Court (SSC) on Wednesday sentenced five defendants in the "Irbid terror cell" case to death by hanging, the Jordan News Agency, Petra, reported.

The SSC also handed 15-year jail sentences to three defendants, while seven were sentenced to 10 years in prison, one sentenced for seven years and four for three years, Petra reported.

The defendants were found guilty of committing terrorist acts that caused deaths, using weapons in terrorist acts, manufacturing explosives with the intention of perpetrating terrorist acts and possessing weapons and ammunition to commit acts of terrorism...

USE classifier

Court hears defence in 'Irbid terror cell' case

AMMAN — The State Security Court (SSC) on Monday heard statements from the defence in the "Irbid terror cell" case, the Jordan News Agency, Petra, reported. Twenty-one defendants are being tried, and the SSC has hired lawyers for those unable to pay for representation.

The defendants are accused of committing terrorist acts that caused deaths, using weapons in terrorist acts, manufacturing explosives to commit terrorist acts and possessing weapons and ammunition to commit terrorist acts.

Other charges include planning to carry out terrorist acts and promoting terrorist ideologies.

B.3 Examples of semi-relevant articles

The following are examples of articles considered semi-relevant. Both were among the top ranked articles by the topic modeling method.

House speaker says Kasasbeh's murder shows IS' true colours

AMMAN — Lower House Speaker Atef Tarawneh said the way pilot Muath Kasasbeh was killed "clearly reveals the nature" of the so-called Islamic State (IS) as a blood-thirsty, hateful group.

"The group is nothing but a band of terrorists and criminals," he said, describing the killing of the pilot as a "cowardly" act.

He called on Jordanians to stand together in support of the Hashemite's leadership and the Jordan Armed Forces-Arab Army in the fight against the IS.

Five suspected IS supporters face trial

AMMAN — The prosecutor general on Sunday referred five suspects to the State Security Court on charges of "promoting terrorist ideology" and "attempting to recruit for a terrorist organisation" in violation of the Anti-Terrorism Law.

The suspects were accused of using the Internet to promote terrorist ideologies of the Islamic State (IS), and trying to join the terror group and encourage others to do join

B.4 Examples of irrelevant articles

The following are the top ranked irrelevant articles found by keyword search and Sentence Transformer, respectively.

Two men sentenced to 6-year prison terms for fraud

AMMAN — The State Security Court on Monday sentenced two Jordanians to six-year prison terms with hard labour and fined them JD600 each for financial fraud.

The court also ruled that the two men pay a JD140,000 guarantee, which represents the amount of money they collected through fraud.

Turkey orders 1,112 arrested over links to cleric Gulen

ISTANBUL — Turkey launched on Tuesday one of its largest operations against alleged supporters of the US-based Muslim cleric accused of leading an attempted coup in 2016, ordering the arrest of 1,112 people, state media reported.

The operation, related to alleged cheating in police examinations, showed authorities were not letting up on their crackdown two-and-a-half years after rogue soldiers used warplanes, helicopters and tanks in a bid to seize power.

More than 250 people were killed in the failed putsch, in which preacher Fethullah Gulen, a former ally of President Recep Tayyip Erdogan, has denied involvement. Gulen has lived in self-imposed exile in Pennsylvania since 1999...

About FFI

The Norwegian Defence Research Establishment (FFI) was founded 11th of April 1946. It is organised as an administrative agency subordinate to the Ministry of Defence.

FFI's mission

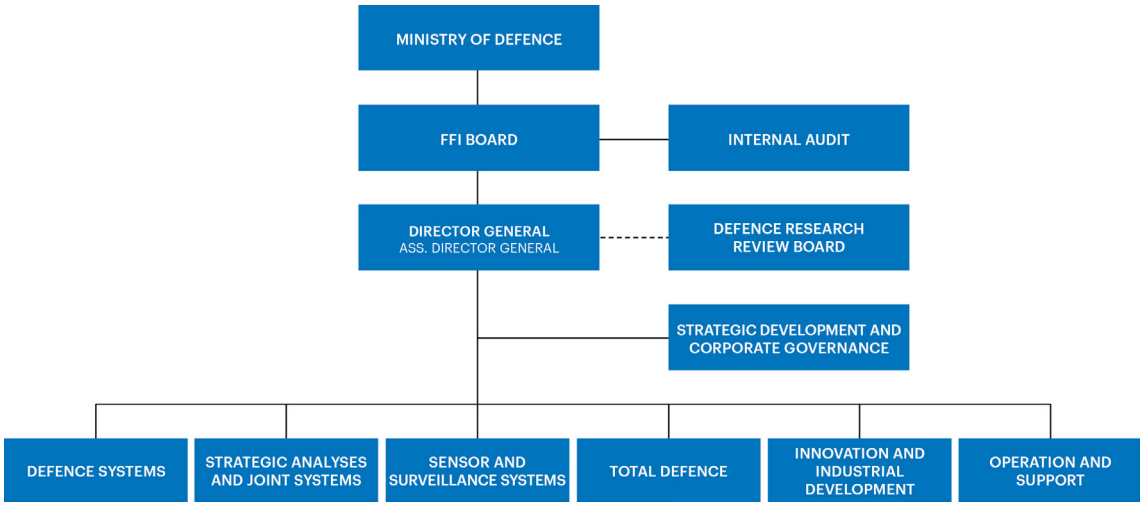
FFI is the prime institution responsible for defence related research in Norway. Its principal mission is to carry out research and development to meet the requirements of the Armed Forces. FFI has the role of chief adviser to the political and military leadership. In particular, the institute shall focus on aspects of the development in science and technology that can influence our security policy or defence planning.

FFI's vision

FFI turns knowledge and ideas into an efficient defence.

FFI's characteristics

Creative, daring, broad-minded and responsible.



Forsvarets forskningsinstitutt
Postboks 25
2027 Kjeller

Besøksadresse:
Instituttveien 20
2007 Kjeller

Telefon: 63 80 70 00
Telefaks: 63 80 71 15
Epost: post@ffi.no

Norwegian Defence Research Establishment (FFI)
P.O. Box 25
NO-2027 Kjeller

Office address:
Instituttveien 20
N-2007 Kjeller

Telephone: +47 63 80 70 00
Telefax: +47 63 80 71 15
Email: post@ffi.no