Textual Analysis of Risk Disclosures

Problem Definition:

Since the introduction of the Capital Asset Pricing Model (CAPM) more than fifty years ago, the asset pricing literature has witnessed a tremendous growth in potential additional factors that could help explain the cross-section of expected stock returns. The formidable economic challenge comes in finding and interpreting economically relevant risk factors. I implemented an approach to this challenge by eliciting the risk factors that firms themselves identify in their annual reports. I then evaluate which ones are systematic, which ones are priced, and whether or not they contain information above and beyond the standard factors and characteristics. To accomplish this, I use machine learning to identify the risks that firms face by applying textual analysis techniques on their annual reports, then I use topic (key words) to represent risk factors.

Data Wrangling:

I use one public source of data: the 10-Ks Annual Reports, 10-K Annual Reports
Firms disclose in their annual reports which types of risk they are facing. There can be some concerns about how true and informative these disclosures are, however, there exist ample evidence that risk disclosure are, indeed, useful and informative. First, firms are legally required to to discuss ”the most significant factors that make the company speculative or risky” (Regulation S–K, Item 305(c), SEC 2005) in a specific section of the 10-K annual reports (Section 1A) and could face legal action if they fail to obey the regulation. Additionally, Campbell et al. (2014) find that ”firms facing greater risk disclose more risk factors”, ”the type of risk the firm faces determines whether it devotes a greater portion of its disclosures towards describing that risk type”, ”managers provide risk factor disclosures that meaningfully reflect the risks they face” and ”the disclosures appear to be specific and useful to investors”.

I extract the textual risk factors in Section 1A (mandatory since 2005) of each 10-K Annual Report. I collect the 10-Ks from 2005 to 2018 from the EDGAR database on the SEC's website. The 10-Ks come in many different file formats (.txt., .xml, and .html) and have different formatting, so it is quite challenging to automatically extract the Section 1A-Risk Factors, from the 10-K forms. I first detect and remove the markup language and then use regular expressions with predefined heuristic rules to extract these sections. I end up with a data set consisting of 37116 documents.

An excerpt of the 10-K annual report of Apple Inc. for the year 2010 illustrates the kind of disclosures that firms make. I incorporate suggested labels regarding the type of risk, as well as highlight possible key words in red. Note that both labels and key words are just for illustrative purposes, and there is no need to manually label the risks in the paper or define the keywords, since the risk factors will arise naturally using the LDA algorithm.

- Currency Risk: ”Demand (...) could differ (...) since the Company generally raises prices on goods and services sold outside the U.S. to offset the effect of the strengthening of the U.S. dollar change”.

- Supplier Risk: ”The Company uses some custom components that are not common to the rest of the personal computer, mobile communication and consumer electronics industries.”

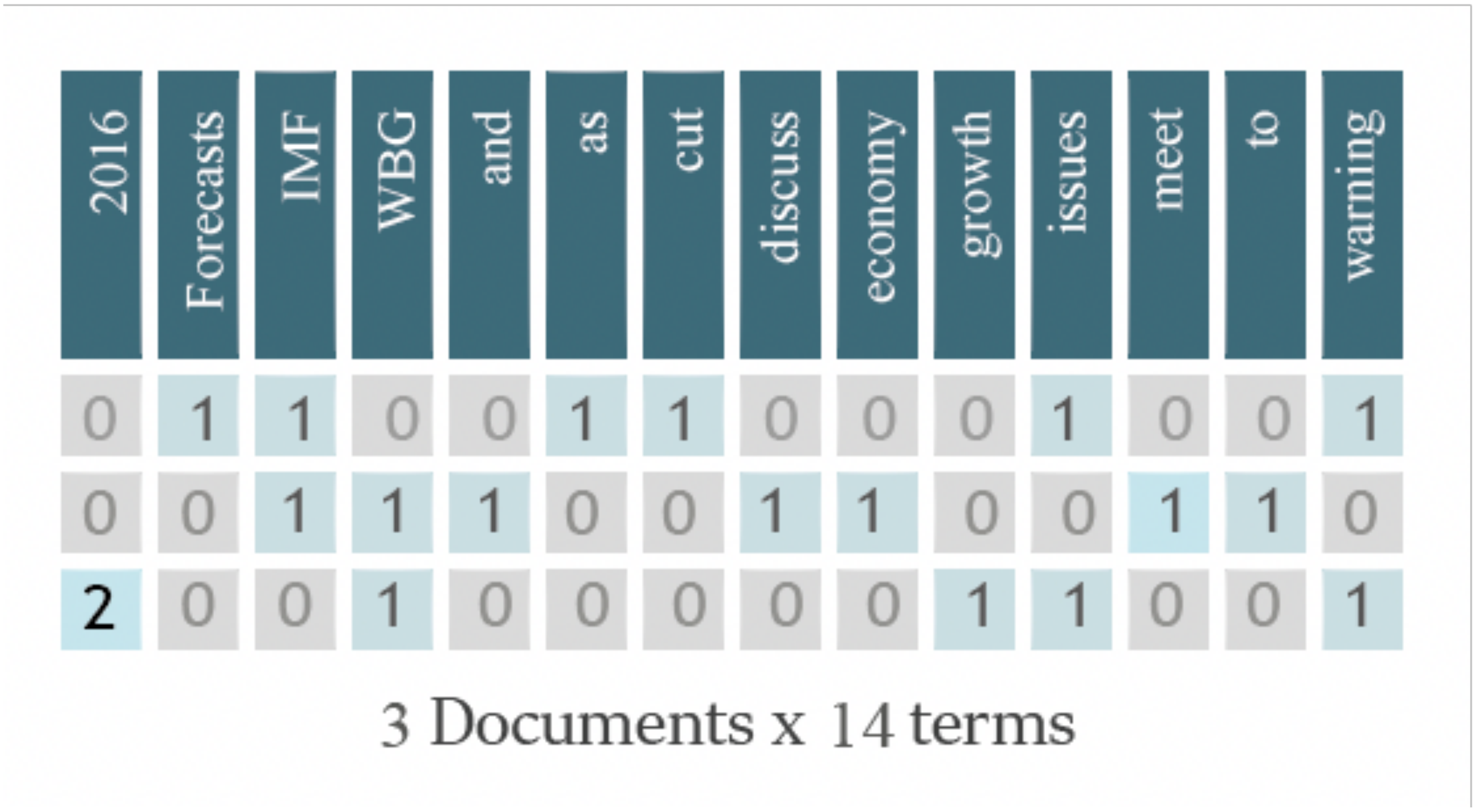
- Competition Risk: ”Due to the highly volatile and competitive nature of the personal com- puter, mobile communication and consumer electronics industries, the Company must con- tinually introduce new products”

Text Processing:

We need a way to represent text data for statistical purposes. The bag-of-words model achieves this task. Bag of Words considers text as a list of distinct words in a document and a word count for each word, which implies that each document is represented as a fixed-length vector with length equal to the vocabulary size. Each dimension of this vector corresponds to the count or occurrence of a word in a document. Traditionally, all words are lowercased to reduce the dimension in half. It is called a "bag" of words, because any information about the order or structure of words in the document is discarded. The model is only concerned with whether known words occur in the document, not where in the document. Notice that since we only consider the count, the order of the words is lost. When we consider several documents at a time, we end up with a Document Term Matrix (DTM), see Figure 2 for a simplified example. The DTM is typically very high dimensional (> 10,000 columns), since we consider the space of all words used across all documents, it is also very sparse, since typically documents do not use the whole English vocabulary. Because of the huge dimension of the space, we need a dimensionality reduction technique, such as LDA.

Another subtle disadvantage is that it breaks multi-word concepts such as "real state" into "real" and "state", which have to be rejoined later, since counting those words separately will be different results than counting the multi-word concept.

Figure 2: Example of a very simple document term matrix



2016	Forecasts	IMF	WBG	and	as	cut	discuss	economy	growth	issues	meet	to	warning
0	1	1	0	0	1	1	0	0	0	1	0	0	1
0	0	1	1	1	0	0	1	1	0	0	1	1	0
2	0	0	1	0	0	0	0	0	1	1	0	0	1

3 Documents x 14 terms

Preprocessing:

It is common to preprocess the raw text in several steps in order to make the topics more interpretable and to reduce the dimension. The purpose is to reduce the vocabulary to a set of terms that are most likely to reveal the underlying content of interest, and thereby facilitate the estimation of more semantically meaningful topics. I remove common English words (the, and, or, ...) and additional terms that do not convey any meaning or are considered legal warnings in the 10-K (materially adverse, no assurance, ...) in order to extract only risk factors from the text.

Some words represent the same underlying concept. For example, ”copy”, ”copied”, and copy- ing; all deal with either a thing made to be similar or identical to another or to make a similar or identical version of. The model might treat them differently, so I strip such words to their core. We can achieve this by either stemming or lemmatization. Stemming and Lemmatization are fundamental text processing methods for text in the English language.

Stemming helps to create groups of words which have similar meanings and works based on a set of rules, such as remove “ing” if words are ending with “ing”. Different types of stemmers are available in standard text processing software such as NLTK (Loper and Bird (2002)), and within the stemmers there are different versions such as PorterStemmer, LancasterStemmer and SnowballStemmer. The disadvantages of stemming is that it cannot relate words which have different forms based on grammatical constructs, for example: ”is”, ”am”, and ”be” come form same root verb, ”be”, but stemming cannot prune them to their common form . Another example: the word better should be resolved to good, but stemmers would fail to do that. With stemming, there is lot of ambiguity which may cause several different concepts to appear related. Axes is both a plural form of axe and axis. By chopping of the ”s”, there is no way to distinguish between the two.

Lemmatization in linguistics is the process of grouping together the inflected forms of a word so they can be analyzed as a single item, identified by the word’s lemma, or dictionary form(Manning, Raghavan, and Schu¨tze (2008)). In order to relate different inflectional forms to their common base form, it uses a knowledge base called WordNet. With the use of this knowledge base, lemma- tization can convert words which have a different form and cannot be solved by stemmers, for example converting “are” to “be”. The disadvantages of lemmatization are that it is slower com- pared to stemming, however, I use lemmatization to preserve meaning and make the topics more understandable.

Phrase Modeling is another useful technique whose purpose is to (re)learn combinations of tokens that together represent meaningful multi-word concepts. We can develop phrase models by looking for words that co-occur (i.e., appear one after another) together much more frequently than you would expect them to by random chance. The formula to determine whether two tokens A and B constitute a phrase is:

$$\frac{count(A,B)-count_{min}}{count(A)*count(B)} * N \geq threshold \text{ , where:}$$

- count(A) is the number of times token A appears in the corpus
- count(B) is the number of times token B appears in the corpus
- count(A, B) is the number of times the tokens A and B appear in the corpus in that order
- N is the total size of the corpus vocabulary
- $count_{min}$ is a parameter to ensure that accepted phrases occur a minimum number of times
- threshold is a parameter to control how strong of a relationship between two tokens the model requires before accepting them as a phrase

With phrase modeling, named entities will become phrases in the model (so new york would become new york). We also would expect multi-word expressions that represent common concepts, but are not named entities (such as real state) to also become phrases in the model.

Dictionary methods:

The most common approach to text analysis in economics relies on dictionary methods, in which the researcher defines a set of words of interest and then computes their counts or frequencies across documents. However, this method has the disadvantage of subjectivity from the researcher perspective since someone has to pick the words. Furthermore, it is very hard to get the full list of words related to one concept and the dictionary methods assume the same importance or weight for every word. Since the purpose of the paper is to extract the risks that managers consider important with minimum researcher input, dictionary methods are unsatisfactory.

Furthermore, dictionary methods have other disadvantages, Hansen, McMahon, and Prat (2018) say: ”For example, to measure economic activity, we might construct a word list which includes ‘growth’. But clearly other words are also used to discuss activity, and choosing these involves numerous subjective judgments. More subtly, ‘growth’ is also used in other contexts, such as in describing wage growth as a factor in inflationary pressures, and accounting for context with dictionary methods is practically very difficult.”

For the purpose of studying the cross-section of returns, the problem is similar to picking which characteristics are important for the returns. The dictionary methods would be equivalent to manually picking which characteristics would enter a regression. The following algorithm, Topic Modelling, is akin to automatic selection methods, such as LASSO (Tibshirani (1996)).

Topic Models:

A topic model is a type of statistical model for discovering a set of topics that describe a collection of documents based on the statistics of the words in each document, and the percentage that each document spends in each topic. Since in this case, the documents are the risk disclosures from the annual statements and they only concern risks, the topics discovered will correspond to different types of risks.

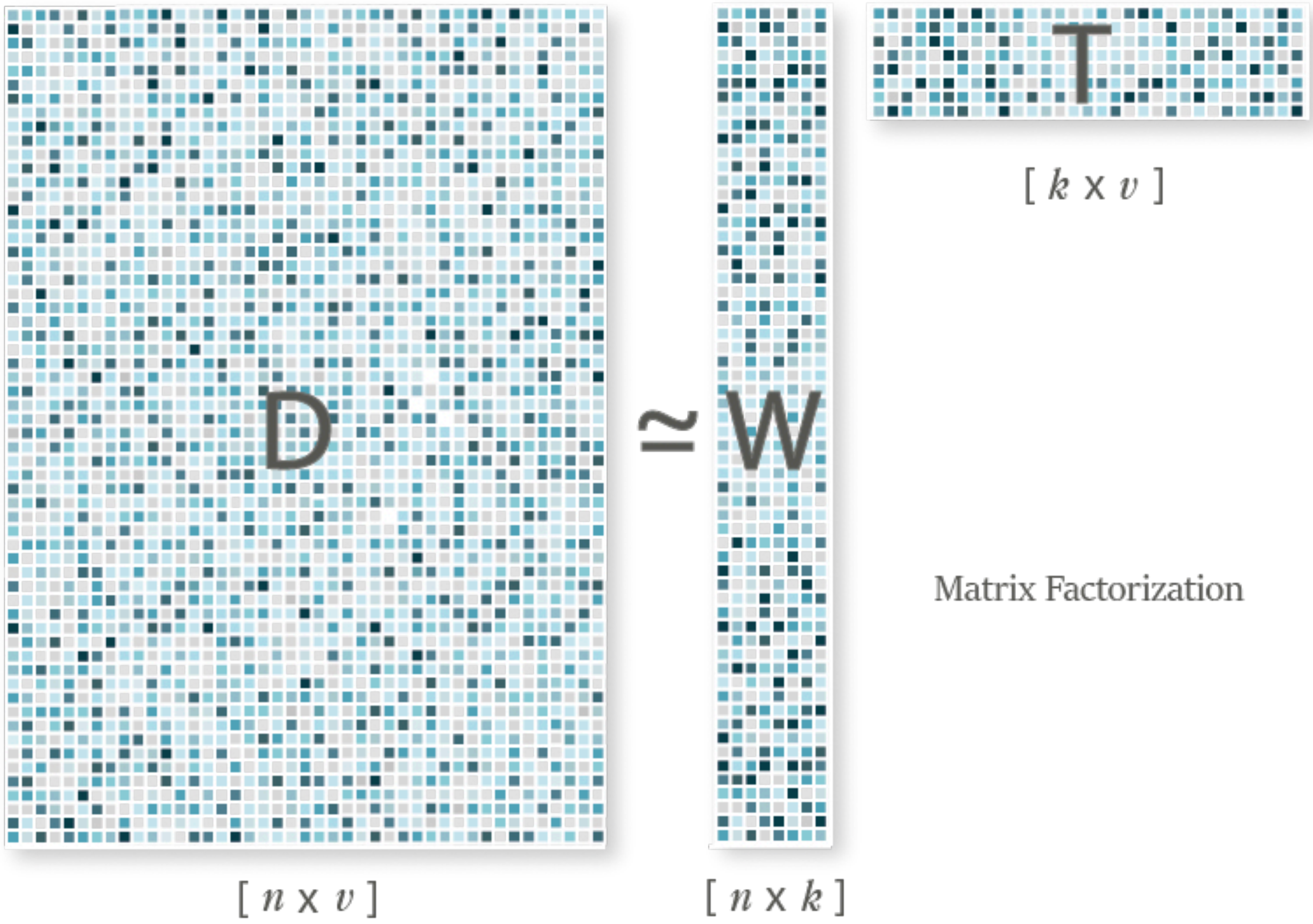
Intuitively, given that a document is about a particular topic, one would expect particular words to appear in the document more or less frequently. For example: ”internet” and ”users” will appear more often in documents produced by firms in the technology sector; ”oil”, ”natural gas” and ”drilling” will appear more frequently in documents produced by firms in the oil industry, while ”company” and ”cash” would appear similarly in both.

A document typically concerns multiple topics, or in this case risks, in different proportions; thus, in a company that is concerned with 20% about financial risks and 20% about internet operations, the risk report would approximately have around 8 times more technology words than financial words.

Because of the large number of firms in the stock market, the amount of time to read, categorize and quantify the risks disclosed by every firm is simply beyond human capacity, but topic models are capable of identifying these risks.

The most common topic model currently in use is the LDA model proposed by Blei, Ng, and Jordan (2003). The model generates automatic summaries of topics in terms of a discrete proba- bility distribution over words for each topic, and further infers per-document discrete distributions over topics. The interaction between the observed documents and the hidden topic structure is manifested in the probabilistic generative process associated with LDA.

Figure 3: Intuition for Topic Modelling



LDA :

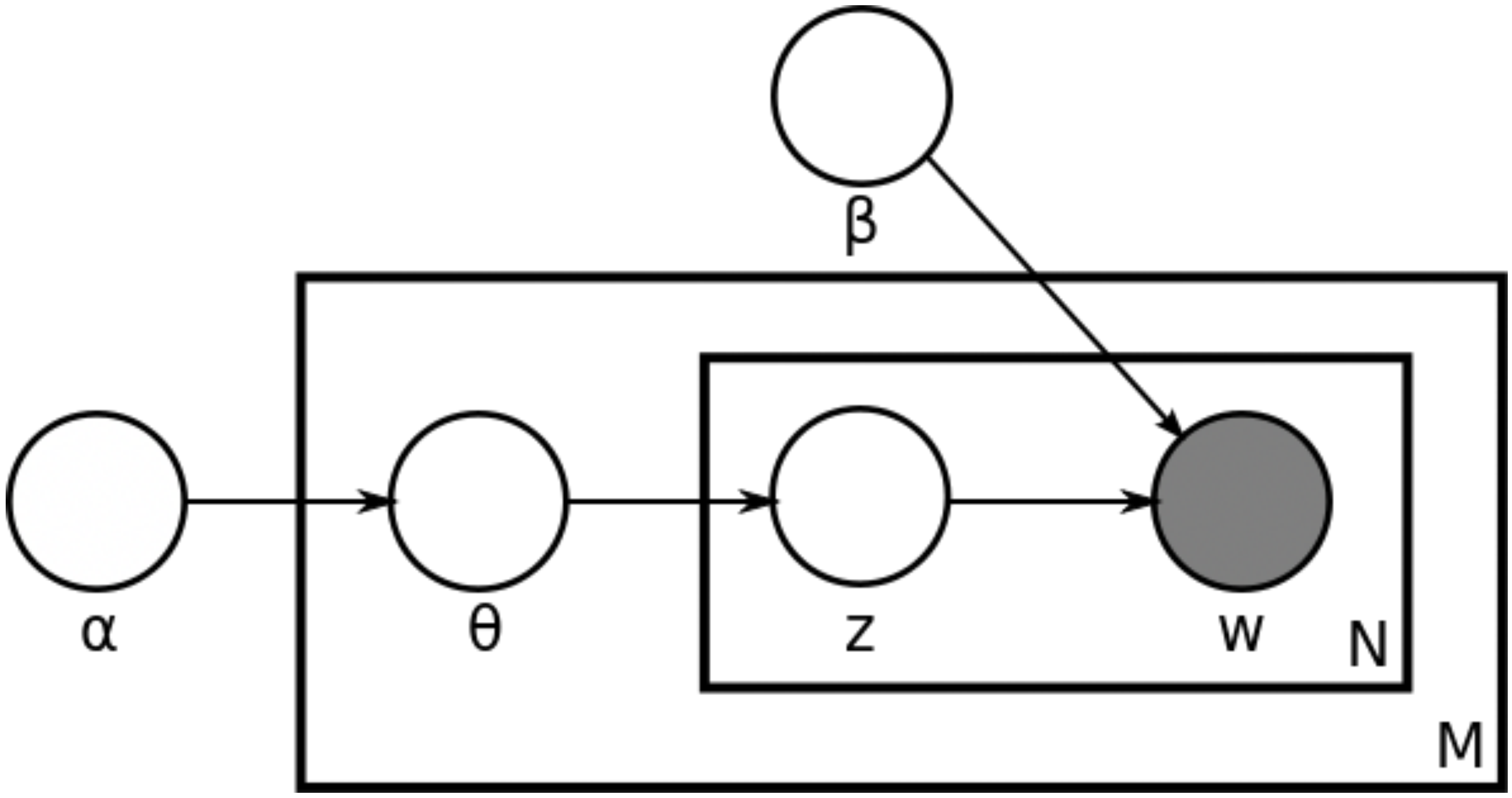
In LDA each document can be described by a (probability) distribution over topics and each topic can be described by a (probability) distribution over words. In matrix algebra terms, we are factorizing the term-document matrix D into a matrix W mapping words to topics, and a matrix T mapping topics to words, similar to the factorization used in Principal Component Analysis, see Figure 3. In this way, LDA reduces the dimentionality of each document, from thousands of words, to the number of topics (20 in our case). However, LDA retains most of the information about the individual word counts, since the topics themselves are probability distribution over words

Formally, LDA is a Bayesian factor model for discrete data that considers a fixed latent set of topics. Suppose there are D documents that comprise a corpus of texts with V unique terms. The K topics (in this case, risk types), are probability vectors $\beta k \in \Delta V -1$ over the V unique terms in the data, where ΔM refers to the M-dimensional simplex. By using probability distributions, we allow the same term to appear in different topics with potentially different weights. We can think of a topic as a weighted word vector that puts higher mass in words that all express the same d has its own distribution over topics given by θd (in our case, how much each company discuss each type of risk). Within a given document, each word is influenced by two factors, the topics underlying theme.

In LDA, each document is described by a distribution over topics it belongs, so each document proportions for that document, θd (in our case, how much each company discuss each type of risk). Within a given document, each word is influenced by two factors, the topics proportions for that document, θ_{dk}, and the probability measure over the words within the topics. Formally, The probability that a word in document d is equal to the nth term is $pdn\theta d$.

It is easier to frame LDA in the language of graphical models, see Figure 4. Where M is the set of all the documents; N is the number of words per document. Inside the rectangle N we see w: the words observed in document i, z: the random topic for jth word for document i, θ: the topic distribution for document i. α: the prior distribution over topics, intuitively controls the sparsity of topics within a document (i.e. how many topics we need to describe a document). β the prior distribution of words within a topic, controls how sparse are the topics in terms of words (i.e. how many words we need to describe a topic). There is a trade-off between the sparsity of the topics, i.e. how specialize they are, and the number of topics.

Figure 4: LDA Graphical Model



The number of topics is a hyperparameter in LDA. Ideally, there should be enough topics to be able to distinguish between themes in the text, but not so many that they lose their interpretability. In this case 20 topics accomplish these task, and is consistent with the numbers used in the literature (Israelsen (2014), Bao and Datta (2014)).

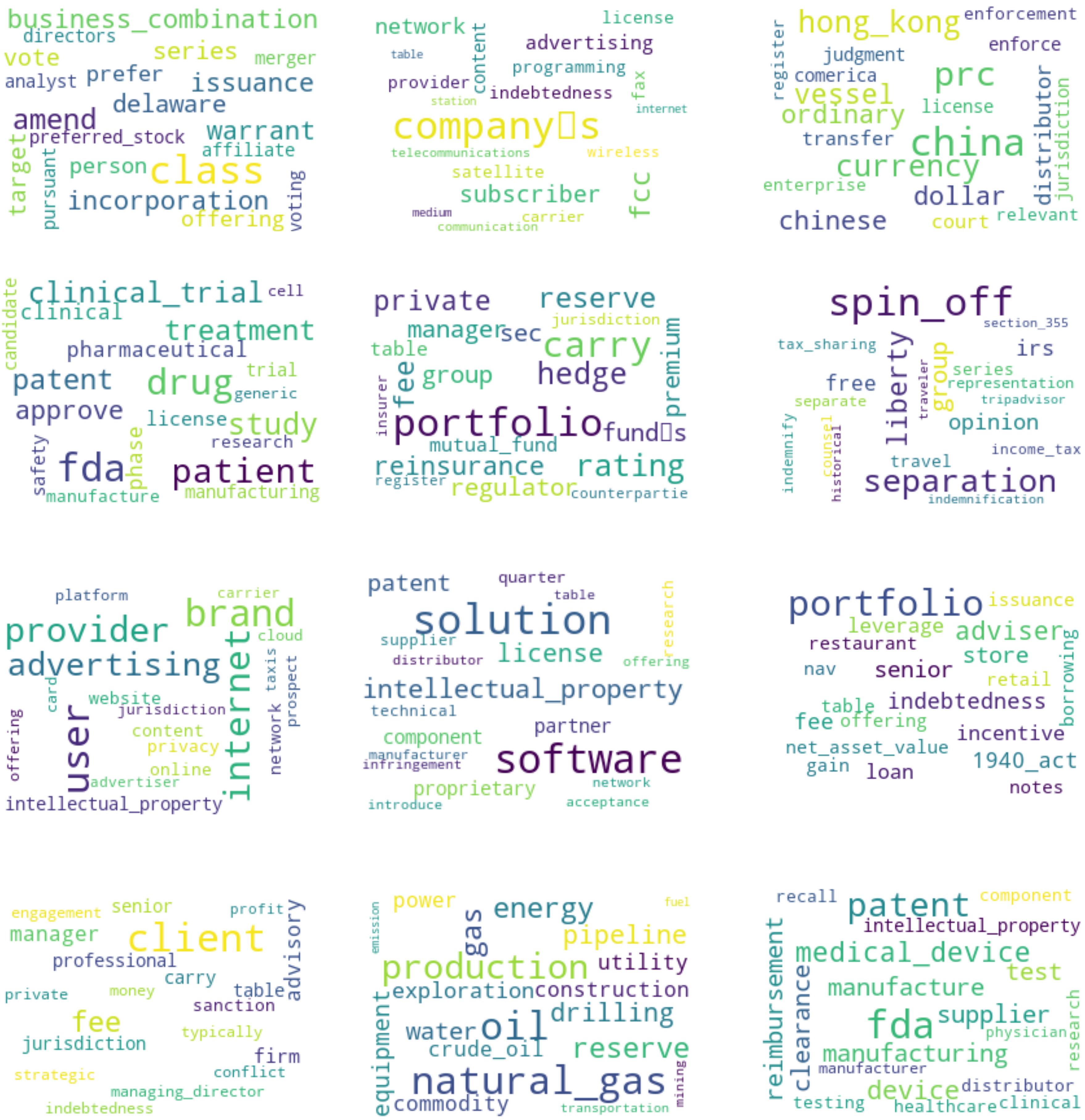
There are technical measures such as perplexity or predictive likelihood to help determine the optimal number of topics from a statistical point of view. These measures are rarely use however, because these metrics are not correlated with human interpretability of the model and prescribe a very high number of topics, whereas for topic models, we care about getting interpretable topics (which correspond to the type of risks).

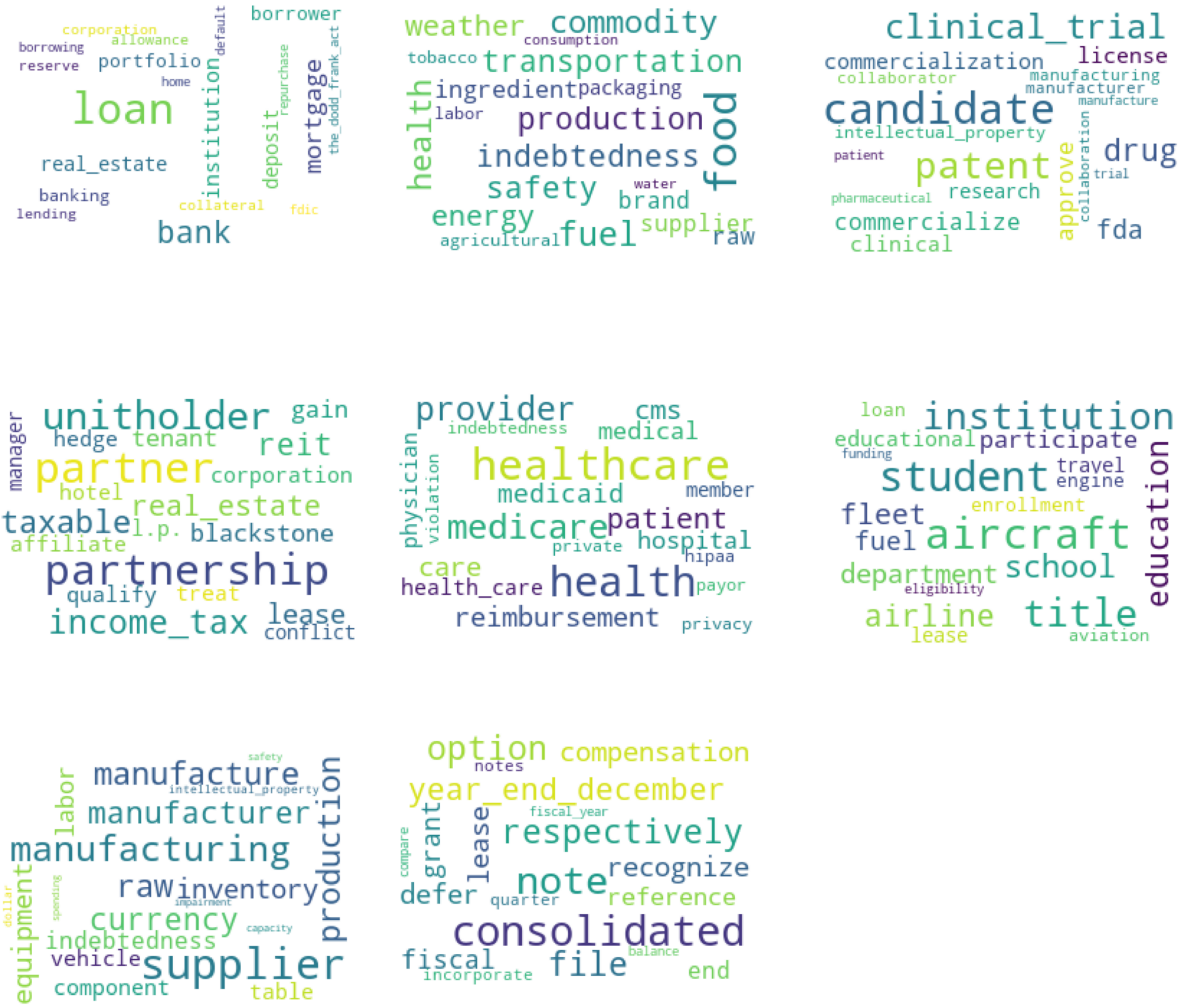
In the case of risk disclosures, a low number (< 10) gets few common risks that all firms in all industries face, and a big number (> 30) starts capturing very specific industry risks. Another issue is that with a big number of topics, very few firms will have significant exposure to each risk, and hence portfolios exposed to some risks will be poorly diversified. I set the number of topics equal to 25 after experimenting with different values.

***Risk Topics and Risk Factors*:**

We can get a general picture of the risks that firms are facing from Figure 5, where I present the 20 risk topics extracted from the 10-K annual report, and recall that a topic is a probability distribution over words.

Figure 5: Risk Topics





Wordcloud of the risks that firms face. A bigger font corresponds to a bigger weight for that word within each topic.

It is important to note that LDA does not give us labels for the topics, but nevertheless the topics are easily interpretable since they are characterized mostly by the most frequent words. Regardless, I name the topics for concreteness.

I refer to the topics as obtained using LDA as Risk Topics and the portfolios formed using these risks as Risk Factors.

Conclusion:

Using machine learning and Natural Language techniques, I introduce factors that unambiguously represent economic risk for the firms, are interpretable, and come directly from the companies. In this way, we can use these key words-instructed topics as risk factors to explain cross-section returns as the classic factor model. Next, I want to dig deeper into practical implementation, to do more research in cross-section returns. That means using these key-word-factors to construct portfolios and test on the historical returns to prove these factors have statistics meaning.