Using Language Models to Classify Innovation and Extract Structured Information about Product Innovation from Unstructured News Stories

Thomas Neil Kattampallil, https://orcid.org/0000-0002-4092-1897, nak3t@virginia.edu
Nathaniel Ratcliff, https://orcid.org/0000-0003-4291-1884
Gizem Korkmaz, https://orcid.org/0000-0002-4947-6320
Alan Wang, https://orcid.org/0000-0001-6926-4336
Steve Zhou, https://orcid.org/0000-0002-1203-7172
Social and Decision Analytics Division, Biocomplexity Institute, University of Virginia

Gary Andersen, NSF
John Jankowski, https://orcid.org/0000-0001-5565-7104
National Center for Science & Engineering Statistics

Social and Decision Analytics Division
Biocomplexity Institute & Initiative
University of Virginia

January 24, 2023

Funding: This research was completed under a contract with the National Science Foundation, National Center for Science and Engineering Statistics Award # 49100420C0015.

Citation: Kattampallil T, Ratcliff N, Korkmaz G, Wang A, Zhou S, Anderson G, Jankowski J, (2023). Using Language Models to Classify Innovation and Extract Structured Information about Product Innovation from Unstructured News Stories, *Proceedings of the Biocomplexity Institute*, Technical Report. TR# BI-2023-5, University of Virginia. https://doi.org/10.18130/hatt-jk83.

UNIVERSITY of VIRGINIA
BIOCOMPLEXITY INSTITUTE

Abstract
Innovation, the availability and usage of novel products and business practices, is central to improving living standards. Policymakers, in part, rely on survey-based measures of innovation to design, develop, and implement policies to promote innovation. In the U.S., the National Center for Science and Engineering Statistics (NCSES) measures innovation through nationally representative surveys of businesses, such as the Annual Business Survey (ABS). To reduce respondent fatigue and to provide more timely information, statistical organizations are interested in exploring non-traditional methods for measuring innovation to supplement existing data.

In this technical report, our goal is to document our research that demonstrates how a large corpus of opportunity data, in particular, news articles, used with advanced natural language processing methods, can be used to identify and measure innovation in various sectors (food and beverage, pharmaceutical, and computer software). We present a novel approach utilizing the Bidirectional Encoder Representation from Transformers (BERT) language model developed by Google. Our methods include (i) text classification to identify news articles that mention innovation, (ii) named-entity recognition (NER), (iii) question answering (QA) to extract company names, and (iv) developing yearly innovation indicators for companies in these sectors.

Keywords—BERT, Natural Language Processing (NLP), Dow Jones, innovation

Table of Contents

Introduction

Innovation is traditionally measured through surveys of selected companies, such as the U.S. Annual Business Survey (ABS) conducted by the National Center for Science and Engineering Statistics (NCSES), a principal statistical agency located within the National Science Foundation, and the European Union Community Innovation Surveys. We focus on product innovation, defined in OECD's Oslo Manual (OECD/Eurostat 2018). as a "new or improved good that differs significantly from the firm's previous goods, and that has been available to potential users." OECD has called for national statistical offices to explore non-traditional methods for measuring innovation outside these traditional surveys (OECD/Eurostat 2018). This technical report uses news articles and natural language processing (NLP) methods, leveraging Google's BERT (Kaput 2021) to measure business innovation. We focus on the pharmaceutical, food, and software sectors due to their high rates of innovation. The three main tasks we accomplished to achieve this goal are:

1. Text classification to identify news articles that mention innovation,
2. Named-entity recognition (NER), and
3. Question answering (QA) to extract company names from these articles to identify the innovators.

The contributions and findings of this report are listed below:

- Developed and optimized a classification model to detect potential innovation articles. *(We use the term "innovation articles" to describe product innovations.)* We use human-labeled data to train this classification model, and we compare the model results to a test set that has been classified by volunteers. We calculate and present the performance metrics, i.e., precision, recall, and f1-score (described in Exhibit 2).
- Implemented multiple iterations of the NER model to identify companies mentioned within the potential innovation articles detected by the classification model.
- Developed a QA model with various degrees of accuracy (see Table II) and used it to extract company names from 20K potential innovation articles detected by the classification model.
- Extracted and identified specific products from innovative companies, a level of detail that earlier methods were unable to achieve.

The motivation for this research is to provide supplementary information to the traditional innovation surveys using alternative data sources. These data sources are unstructured text articles published in business journals and news websites. We accessed these articles from various publishers and sources. They include content released by companies to draw attention to their new product releases and articles developed by journalists who report on business news such as business policies, financial news, industry trends, etc.

One of the objectives is to find more granular and detailed information on innovation, such as the company name, product names, and product features. Additionally, the number of innovative products a company develops each year could be a good indicator of its impact on the industry and a growth indicator for that company. To extract this data from unstructured articles, it is necessary to build text processing and NLP systems capable of recognizing text context and identifying entities mentioned in an article. Additionally, using a text processing based approach allows us to obtain near real-time estimates on innovation in the economy.

Related Work

Innovation measurement through text

Using several different datasets and NLP techniques, we explored the innovation measurement domain through text. One such example is by Bellstam, Bhagat, & Cookson (2021). This project uses Latent Dirichlet Allocation (LDA) to identify innovation texts in a dataset of analyst reports of companies in the S&P 500, similar to the company news articles we use in our analysis. This research does show that finding innovation indicators from text data is a viable strategy.

Another example is using Web Content (Héroux-Vaillancourt, Beaudry, & Rietsch 2020). The authors scraped web content to build an innovation measurement index based on keyword searches. In the paper, they discuss their process of scraping websites of companies, to obtain the text content, and then looking for specific keywords in this text that could indicate innovation. The main problem lies in the need for more context tied to using keywords alone, possibly leading to multiple false positives.

Machine learning and deep learning techniques, such as recurrent neural networks, natural language processing, or bag-of-words models, are promising avenues to explore the context surrounding specific concepts to improve the precision of web-based indicators (Vaswani et al. 2017). These are explored below.

Our methods build on these suggested approaches to build models that can identify innovation more precisely. Additionally, this paper focuses on the text from corporate websites as a data source. It is restricted to nanotechnology and advanced materials in Canada.

Another example is “Predicting innovative firms using web mining and deep learning” (Kinne & Lenz, 2021). In this paper, the authors aim to build an innovation indicator for policymaking. The dataset that they use is the web texts from firms in Germany. They build an artificial neural network to classify these web texts as ‘product innovator/no product innovator’. They compare these model results to the Mannheim Innovation Panel [MIP], an annual questionnaire-based innovation survey of firms sampled from the Mannheim Enterprise Panel (MUP) database. (The MUP is a panel database covering the total population of firms in Germany). They were able to achieve a precision score of 0.72 for non-innovative firms and 0.62 for innovative firms, suggesting that, for their chosen thresholds, their models were able to use the text from company web pages to decide if the company was non-innovative, but there was comparatively lower precision when deciding that a company was innovative. Furthermore, they obtained a recall score of 0.83 for non-innovative but only 0.47 for innovative, which means that the model was able to capture a large proportion of non-innovative articles, but captured a much smaller proportion of the innovative articles

Kinne and Lenz (2021) research is interesting because they also apply neural networks to identify innovative firms from text data and compare their results to a survey designed according to the Oslo Manual definition of innovation. In our work, we also use machine learning to identify innovation in the article text to supplement the results obtained from a survey that follows the Oslo Manual definition of innovation.

Research using BERT for unstructured text

Natural Language Processing (NLP) has seen significant advances through Google BERT. BERT, which stands for Bidirectional Encoder Representations from Transformers, is a novel method for language models in that it trains and encodes words based on the context of words both to the left and right of each word in a sentence (Kaput, 2021)

Since BERT was released, many experts have started applying it to their field of expertise. Researchers at Korea University in Seoul developed BioBERT (Lee et al. 2019) and trained it on a combination of four corpora in Named-Entity Recognition (NER) and Question Answering (QA). These included the entirety of English Wikipedia, PubMed abstracts, PubMed full articles, and BooksCorpus. They tested this trained NER model using the GAD (Gene-Disease Associations) Dataset). The GAD contains text snippets of descriptions of gene-disease associations curated from genetic association studies. BioBERT was used to extract Gene–disease entity pairs from this text and was able to do so with an f1 score of 84%, with a recall of 91%, a precision of 78%, and a specificity of 71% (see exhibit 2 for metric definitions). Including PubMed, corpora improved results across all tests. The results were improved by including articles similar to those researchers would use in testing. QA results were lower, with accuracy topping out at 57%, but adding PubMed improved accuracy (Lee et al. 2019).

Other researchers found that BERT could be fine-tuned to cover different disciplines (Beltagy, Lo, & Cohen 2019). As a result, they created a model called SciBERT that is pre-trained on scientific text. Some of their tasks included NER and text classification that relied on a corpus of computer science literature and biomedical literature. They found BERT helpful in both corpora. In the case of the biomedical corpus, researchers obtained as high as 90% accuracy in NER. Compared to BioBERT, SciBERT further improved accuracy in testing on the same corpora. In text classification, their model obtained an accuracy of 85% in a separate corpus (Beltagy, Lo, & Cohen 2019).

Additionally, there have been extensive attempts to improve BERT models by increasing the size of the training set used in initial pre-training. RoBERTa is one such model that iterates on BERT's pre-training procedure. This model includes training the model longer, with bigger batches over more data; removing the next sentence prediction objective; training on longer sequences; and changing the masking pattern applied to the training data.

There have also been attempts to make smaller, lightweight language models using the same BERT-like approach. DistilBERT is one such model that uses knowledge distillation during the pre-training phase to create a smaller, faster, and lighter model which is cheaper to pre-train.

Labeling

Why do we have to Label text?

To build a classification system for the articles, labeling is a necessary first step to obtaining "true innovation" articles. These articles reliably indicate innovation, as well as 'false innovation' articles known not to contain innovation-relevant information. We need these 'true' and 'false' articles to serve as the outcome for the article data to build models for predicting innovation. This mechanism allows us to capture human expertise and feed it to a model to train it. Put simply, many machine learning techniques rely on an outcome to build the model. To have an outcome, we need to ask the same question to a human being and get their decision to build a training set with labeled data.

Labeling Approaches

Automated Labeling

Using existing Machine Learning (ML) and Language models to generate labels for articles without requiring a large human-labeled set is possible. This approach uses semantic similarity, topic similarity, and a handful of example articles to detect articles with associated topics and ideas like those in the example articles. Some of the techniques for this kind of labeling include the use of One-Shot and Zero-Shot Neural networks. These systems aim to make predictions for an NLP task without seeing even one

labeled item (for Zero shot learning) or very few labeled articles (One-shot learning). Researchers can implement these systems using various NLP libraries such as Flair and its TARS classifier, short for Text Aware Representation of Sentences.

There are advantages to using ML and Language mores for generating labels. They include the following:

- Allows for labeling all valid articles
- Can be more accurate than human labeling if accurately constructed
- May be necessary to build a novel application and could be a significant step forward
- Flexible and adaptable to changes (can repeat more than a single sample labeling exercise)

There are disadvantages to this approach as well:

- The method proposed is relatively new and not extensively verified as a "best practice" labeling methodology
- Without manual labeling verification, it is impossible to validate the 'true' accuracy of the labeling model
- As an unsupervised method, there is little to no information about the inner workings of the model

Manual Labeling:

Mechanical Turk

As an alternative to internally labeling the articles, we have designed an instructional guide, called a survey, containing the articles and instructions for answering questions which will give us labels, keywords, company identification, etc. We can then outsource this survey to a service through Amazon called Mechanical Turk (MTurk). This online platform posts tasks, in our case, the task of answering our survey**,** and offers them up to participants, commonly called "Turkers," who self-select and complete tasks for a specific amount of money per task. This approach is a convenient workaround to obtain labeled articles without having to do it ourselves. However, there are three challenges associated with this method:

1. Designing an effective survey is complex, requiring lots of planning, testing, reliability analysis, etc. This task is both time consuming and may only lead to accurate results or labels if we adequately construct the survey.
2. Given the first issue, the cost of outsourcing the labeling process increases very quickly. As a first run, labeling 1000 articles by this method maybe cost-effective. (The cost for labeling on Mechanical Turk is based on several factors, including the complexity of the task and educational requirements of the labelers.) MTurk payment must be at least minimum wage. Then a multiplier is added to account for education, location, and language (Amazon MTurk 2022). There is also an additional 20% fee that goes to Amazon as payment for the use of the platform). Thus, MTurk can be more expensive than internal labeling. (As of January 1st 2022, minimum wage in the state of Virginia is $11 an hour, and from our experiences in volunteer-driven labeling exercises, it takes people around an hour to do 15 articles; this average varies based on topic, we noted that food related articles were a lot faster to label, while pharmaceutical related articles took longer, reflective of the relative complexity of the texts. This gets us to a cost of $733 for the Turkers, and with the additional 20% fee from Amazon, the total is around $880 for 1000 articles, without taking into consideration details such as the level of education that we expect from the MTurk workers, and the changes to the survey that need to be made to ensure that the workers are not labeling articles without reading them)

3. Given the first two issues, we must generate a sample of articles to be sent off and labeled. Based on the OSLO Manual definition, we began with 2 million articles to filter down to about 150,000 or fewer valid innovation articles per year. Stratified sampling is necessary to ensure the articles are representative of the target population and that future models, based on these labels, are generalizable and valuable to our specific aim.

Expert Labeling

Another possible approach is to extract relevant data from the articles' data set, such as an article's title and body content, and other metadata, such as publisher name. We then present these data to someone familiar with the Oslo Manual definition of innovation. They read through the article and decide whether it is about product innovation. To this end, we have developed an instructional document (Ratcliff & Kattampalil, 2022) to familiarize people with the Oslo definition of product innovation and highlight complex cases for a person to decide whether the article is really about product innovation. This instructional document guides the labeler's decision-making process; thus, we expect this to be the most accurate way to label the articles. Once trained, we have found that it takes students about half an hour to label 20 articles. However, this can be taxing in terms of time and limits the total number of articles to label and use to build the initial models.

Data

We discovered data sources for three industries studied in our research: Pharmaceutical industry (Drugs and Medical Devices), Food and Beverages, and Software Development industry. The Food & Drug Administration (FDA) regulates the pharmaceutical industry (FDA 2022a; FDA, 2022b). This regulation allows us to use publicly available FDA databases to identify companies producing new products. Through this validation process, we can identify whether our methods work for that specific industry and whether we can apply it identically to other industries that are regulated. Additionally, these three sectors were selected as they display higher than average levels of innovation compared to other sectors in 2017-2019 (NSF NCSES 2022).

- All industries. 10.9%
- Food. 13.2%
- Pharma 24.3%
- Software. 43.6%

Dow Jones – DNA Dataset

We discovered and acquired the dataset of articles from Dow Jones Data, News, and Analytics (DNA). The Dow Jones DNA platform collects information from Dow Jones publications from premium and licensed third-party sources. This proprietary data platform contains 1.3 billion articles labeled with unique DNA taxonomy tags, including word count, source name, and company code. The dataset includes variables such as the publisher, publication date, and companies mentioned (codes and names) in each article.

DNA provides metadata about articles, which is generated through a combination of human labels and proprietary machine-learning algorithms implemented by Dow Jones/DNA internally. As a result, we cannot use the metadata tags as a reliable ground truth. Still, we can use them to help in our article selection process when building a training dataset for our machine-learning models. For example, the C22 code, the metadata tag for ‘new products and services,’ was helpful, as those articles tended to be about product launches, which had a high likelihood of being innovation related.

The data used in this project consists of the following:

- 1.8 million news articles about the Pharmaceutical Industry published between 2013 and 2018,
- 0.6 million news articles about Food Processing and Beverage Manufacturing published between 2013 and 2021, and
- 1.2 million articles about Computer Systems Design and Software development, published between 2013 and 2021.

Other Data Sources Explored
We explored SEC Filings and Patent data.

Patent information was a potentially interesting source, as it is a well-documented and publicly available dataset that mentions detailed information about each invention and its features. However, patents do not always translate to products available on the market. Patents can be used to restrict other firms from developing competing products, even if there is a demand for them. For example, Hewlett-Packard holds the "Smart Mirrors" patent even though they do not produce or sell a consumer product that uses the technology described within the patent. Additionally, a new product may differentiate itself in the market through design refinements that may not be patentable. Therefore, patents do not fully reflect innovation as per the Oslo definition.

SEC Filings is another promising avenue for information about product innovations in an industry. All publicly traded companies must file annual and quarterly reports. These reports often discuss details of new product releases and revenue earned from these new product launches. While the SEC filings are publicly available, they use a proprietary XML format, which makes it difficult for automated systems to retrieve the information.

Text Representation for Machine Learning
We first translate the articles to numerical data before using them in Machine Learning models. This translation is necessary because the statistical and numerical methods that form the core of all ML models require numerical data to work correctly. There are various methods commonly used to convert text data into a numerical format, and those techniques are known as Text Representations. Some popular methods include Discrete text representation methods like Bag of Words and Continuous Text Representation methods such as word embeddings or word2vec.

Why does Text need to be reformatted for ML?
Bag-Of-Words Method
The Bag of Words model uses both Image Classification and Natural Language Processing. This model is relatively easy to implement and theoretically straight-forward model. After removing stop words (for example, 'the,' 'and,' 'a,' etc.), we use the first article's text body. From there, we tokenize each word found in the document. If a word shows up twice, we associate a two count with that word. If it only shows up once, we give it a 1, and 0 if it does not appear. We now have what's called a bag of words. This tokenization reduces an article's text body into a matrix of individual word keys and associated counts (a long vector of numbers). Then, we repeat this process for all documents in the data, combining these together to get a large matrix with observations being keywords and columns being the documents (or vice versa). We then perform one of a variety of clustering algorithms, semi-supervised machine learning to utilize the underlying structure of the data to classify portions of it (e.g., 2,3, 4,). These are then analyzed to decide which split is associated with the outcome of interest.
For our purposes, we generated a DNA bag of words and clustered it into two groups, one describing innovation and the other not. As mentioned above, we can extend this further to more ordered groups.

There are advantages to using this model:

- it is relatively easy computationally,
- interpretable in a sense (because it is semi-supervised), and
- can be repeated many times to validate results (sensitivity analysis, diagnostics, etc.).

There are also drawbacks. Since we are counting word frequency, this model does not consider the order or relationship between words and sentences. This approach is a drawback if the keywords describing innovation are longer phrases (words consistently occurring in a specific order) rather than individual words.

TF-IDF Method

Term Frequency (TF) — Inverse Dense Frequency (IDF)) is a technique used to determine the relative importance of words and phrases in documents. It overcomes the limitations of the Bag of Words technique by introducing a critical idea called inverse document frequency.

Inverse Document Frequency is a score that the computer maintains when it assesses the words used in a sentence and compares their usage to those used throughout the document. The score emphasizes the importance of each phrase throughout the entire piece of text. The equation is:

IDF = Log[(# of documents) / (documents with the word)]
TF= (# of occurrences of a word in a document) / (# of words in a document)

Term Frequency indicates the number of times a specific word is used in an entire document. IDF indicates the importance of a particular word across all documents in the set. IDF allows us to identify important words and the overall theme of the documents.

Word2Vec

A new development in NLP is called Word2Vec, which is used to produce "Word Embeddings." These play a crucial role in resolving various NLP issues. These word embeddings show a machine how people interpret words. For example, once words are represented in a vector space, similar words are clustered together and can be used with vector arithmetic. Perhaps one of the most famous examples of word2vec arithmetic is the expression king - man + woman = queen. In this way, word embeddings provide information about how human beings interpret the meaning of words so that it can be incorporated into a vector representation. Word Embeddings can be thought of as text that has been vectorized. Well-developed vectorizations can be used in several applications, such as sentiment analysis, recommendation systems, and text similarity.

Word Embeddings convert each word into a numerical representation (a vector). A neural network maps each word to a single vector (word embedding). This set of vectors is subsequently utilized to describe each document as a set of vectors. The vectors attempt to depict various aspects of that word in relation to the whole text. These traits may include the word's semantic relationship, definitions, context, etc. You can determine word similarity or dissimilarity using these numerical representations, among other things.

These are unquestionably essential as inputs for a variety of machine-learning processes. A first step is to convert text into an embedding because a machine cannot process it in its raw form. The embeddings can then be used in conventional machine learning models.

Tokenization using methods like BERT

BERT is the application of Transformer architecture to the area of text analysis. A transformer is a deep learning model that uses a mechanism called 'self-attention' to process sequential data streams, which allows each input part to be assigned a different weight or significance. This 'attention' based weighting allows for additional context for each piece of the input sequence. By feeding a large amount of text input to these transformer models, it is possible to train a model that can recognize streams of text, establish context and form embeddings similar to word2vec, except that it also uses the self-attention mechanism to build the vector representation of words.

BERT has the added advantage of also having sentence-level embeddings (The same word used in different sentences can have different contexts, and the embeddings of the word will be different to reflect that). BERT is also bi-directional, taking the attention values of surrounding words into account when generating an embedding. Through pretraining on a corpus that includes the entirety of Wikipedia and BookCorpus, the BERT pretrained model has an extensive set of embeddings, and the BERT tokenizer is a function that takes plain text and breaks it into words and word pieces that are contained in the BERT pretrained embedding set.

Methods

Innovation Classification

This section discusses classifying articles as innovation or non-innovation based on the article text using Machine Learning.

There are two approaches to address the challenge of having too many examples of one class in a training set, which would cause the model to be biased.

- Subsampling is used to reduce the number of examples of the unwanted class in the training set, in our case, reducing the number of non-innovation articles.
- Supersampling is used to introduce more examples of a specific class in the training set, increasing our case's number of true positives.

We select our training set using metadata from the DNA data set. The DNA code 'c22', which refers to new products, is a piece of metadata that allows us to improve our chances of finding a true innovation article in the training set.

Once the selected articles are labeled, we ensure that the number of accurate innovation articles and non-innovative articles in the training set are equal to achieve a class-balanced data set.

Training BERT

Once we have a training set built with labels, we use this set to train a BERT classification model. BERT has a set of weights for each token[1] in its language model. Using this labeled data set, we can change the neural network weights that take in BERT tokenized articles and outputs whether it is an innovation article.

[1] "BERT uses what is called a WordPiece tokenizer. It works by splitting words either into the full forms (e.g., **one word becomes one token) or into word pieces — where one word can be broken into multiple tokens**. An example of where this can be useful is where we have multiple forms of words." https://towardsdatascience.com/how-to-build-a-wordpiece-tokenizer-for-bert-f505d97dddbb

The labeled data set can then be used to build a classifier. There are multiple approaches to build a classification model, but the process we have used here is a Feature-based approach that consists of the following steps.

1 Download a pre-trained BERT model.
2 Use BERT to turn natural language sentences into a vector representation.
3 Feed the pre-trained vector representations into a model for a downstream task (such as text classification) using a labeled data set for task training fed into a classifier model, which could be a Logistic Regression Classifier, a Support Vector Classifier, or any of the other standardized classifiers available in the Scikit-learn Python package.

There are several viable options for both the BERT language model and classifier used. Several variations of neural networks are available, and there are also many BERT-like models and variations that better suit specific language-based tasks.

Choosing the Right Language Model and Classifier

For each task and each data set, it is essential to pick the best language model to improve performance. The following is a set of test runs with the same data set to select the best language model and neural network classifier kernel that works for that specific task and data set to find innovation in Food and Beverage articles. These are the top 10 best-performing models for our task and dataset, but this can be a very time-consuming process, considering that there are new language models released every few months that are good candidates for these applications. We ran 25 model-kernel pairs and selected the best model based on maximizing true positives and minimizing false negatives. The best model, in this case, was DistilBERT-base, which is not the state-of-the-art model, but performed better in this specific application. (RoBERTa large was state-of-the-art at the time). **Exhibit 1** presents the top ten best-performing models sorted by AUC value.

Exhibit 1: Definitions of measures to determine ML language and classifier combination

Measure	Definitions
accuracy	Number of Correct Predictions/Total Number of Predictions
precision	Precision (also called positive predictive value) is the fraction of relevant instances among the retrieved instances. (True Positives/True Positives+False Positives)
recall	Recall (also known as sensitivity) is the fraction of relevant instances that were retrieved (True Positives/True Positives+False Negatives)
f1_score	The F1-score combines the precision and recall of a classifier into a single metric by taking their harmonic mean. It is primarily used to compare the performance of two classifiers.
auc	AUC, or AUROC (Area under the Receiver Operating Characteristic Curve)*
true_positive	In a binary classifier, the number of observations in a test set classified by the model as True, which are actually True
true_negative	In a binary classifier, the number of observations in a test set classified by the model as False, which are actually False
false_positive	In a binary classifier, the number of observations in a test set classified by the model as True, which are actually False

Exhibit 1: Definitions of measures to determine ML language and classifier combination	
false_negative	In a binary classifier, the number of observations in a test set classified by the model as False, which are actually True
specificity	1-Recall (Used mainly in medical and biomedical fields)

Exhibit 2. Food and Beverage articles: Top 10 best-performing models (based on AUC value) using DistilBERT and RoBERTa and neural network classifier kernels

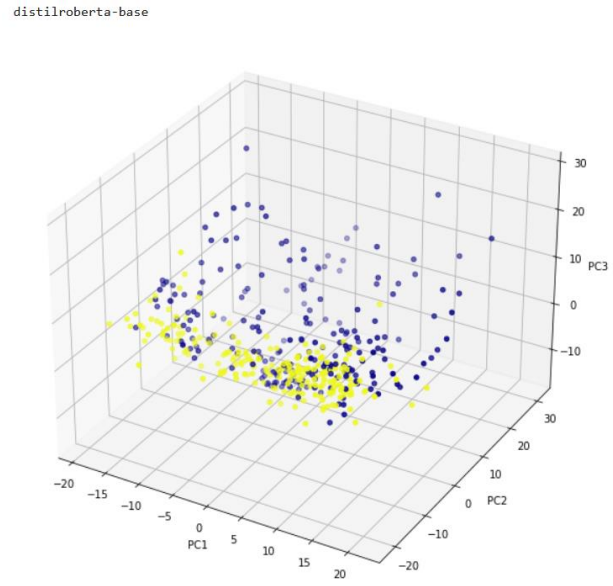
	accuracy	precision	recall	f1_score	auc	true_positive	true_negative	false_positive	false_negative
distilbert-base-cased_rbf-kernel	0.84	0.78	0.95	0.86	0.94	54	39	15	3
roberta-large_rbf-kernel	0.82	0.78	0.89	0.84	0.93	51	40	14	6
distilroberta-base_rbf-kernel	0.85	0.79	0.95	0.86	0.93	54	40	14	3
distilroberta-base_sigmoid-kernel	0.86	0.82	0.95	0.88	0.92	54	42	12	3
distilbert-base-cased_sigmoid-kernel	0.84	0.79	0.93	0.85	0.91	53	40	14	4
distilbert-base-cased_poly-kernel	0.68	0.62	1.00	0.77	0.91	57	19	35	0

Exhibit 2. Food and Beverage articles: Top 10 best-performing models (based on AUC value) using DistilBERT and RoBERTa and neural network classifier kernels									
	accuracy	precision	recall	f1_score	auc	true_positive	true_negative	false_positive	false_negative
distilroberta-base_poly-kernel	0.75	0.67	1.00	0.80	0.90	57	26	28	0
roberta-large_linear-kernel	0.81	0.81	0.82	0.82	0.90	47	43	11	10
roberta-base_rbf-kernel	0.82	0.78	0.89	0.84	0.89	51	40	14	6

To compare model performance, we generate an ROC curve (receiver operating characteristic curve), which is a graph showing the performance of a classification model at all classification thresholds. This curve plots two parameters, True Positive Rate (Recall) vs False Positive Rate at different classification thresholds. We then calculate the 'AUC', or Area under the ROC Curve, which is an aggregate measure of performance across all possible thresholds for classification and a good measure of comparison between models. **Exhibit 2** provides definitions of measures to determine ML language and classifier combination to determine best performance.

To provide a clear decision boundary between the articles we want to classify, selecting these parameters correctly is essential. The BERT model is a high dimensional vector, and is difficult to visualize, therefore one approach is to use Principal Components Analysis to obtain the first 3 principal components of the vector to provide a human-readable visualization. The following illustrates the first three principal components to provide a visualization **(see Exhibit 3).**

Exhibit 3: Scatterplot to visualize the text articles in a vector space, with non-innovative articles in blue, and innovative articles in yellow.



Hyperparameter Tuning

In general, the reason we use pre-trained language models is because training an NLP model from scratch is time consuming and expensive. The process we use, of feature-based classification, takes advantage of the language models tokenization methods, which can convert plain text into a vector representation that preserves semantic relationships and context, which can be fed into a classifier for a specific task.

Another possible option to build a classifier is to use fine-tuning on the pre-trained NLP model to obtain an NLP model optimized for a specific task. This fine-tuning process depends on several variables that we need to select to obtain a correctly optimized model. The variables or parameters that control the learning process of an NLP model are called Hyperparameters. These values are set by the programmer building the models, and the speed and quality of the learning process depends on them. The Hyperparameters depend on the classification model used. For example, if it is a decision tree model, the number of branches in the tree could be a parameter. If it is a clustering algorithm like K-means clustering, the number of clusters is a Hyperparameter.

In Neural networks and transformers like BERT, the Learning rate is an important Hyperparameter, and it is the value by which weights in a Neural Network are adjusted in each cycle of training. Choosing a large value for Learning rate can make the training process much faster but could also result in a final model with lower accuracy. Conversely, choosing a smaller value can result in a more finely tuned model, but could be too time consuming to train. A popular option is to perform a Grid search to choose the correct values for these parameters. The drawback is that while a single training cycle for fine-tuning is relatively quick, having to repeat this process with different hyperparameter values to find the perfect configuration is time consuming.

Additionally, especially in the case of relatively small training datasets, trying to optimize and tune parameters can backfire, even with a small learning rate and a large number of training cycles (also called ‘epochs’) because it can result in overfitting, which is where the model becomes so finely tuned to the patterns in the training data that it expects the same patterns in real-world data, which does not occur, and this reduces the overall accuracy of the model. For this reason, we have stuck to using a feature-based classification model for our tasks.

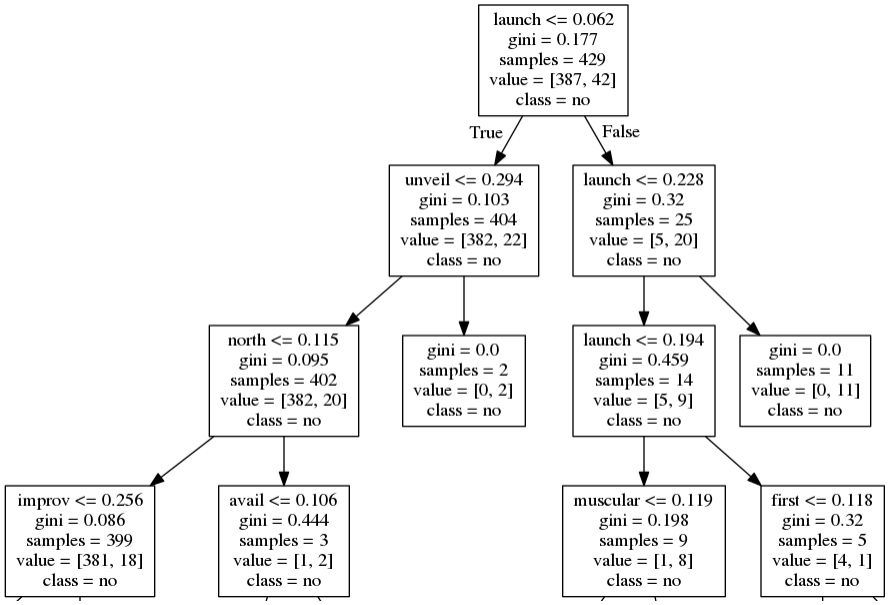
Based on the initial analysis of the pharmaceutical sector, the word 'launch' was indicative of an article describing an innovative product. When working in a tf-idf representation of the text, the presence of a specific word in an article, such as 'launch,' is the main feature that the model uses to make predictions. This pattern provides a naïve baseline method. Suppose the models we develop could detect true innovation articles better than a simple model that searches for the word 'launch.' In that case, we can objectively measure the performance of our methods. See Exhibit 4. For an example of a decision tree based classifier that we had initially developed.

When initially exploring datasets, we had used the TF-IDF text representation, so each document was represented as a vector of values, where each value represented the TF-IDF value of a certain word. This allowed us to start to measure which words are the most important indicators of innovation. In practice, this can be an oversimplification; the context of word usage can be just as important as the presence or absence or probability of occurrence of the actual word. This initial analysis was simply to analyze whether there were any obvious patterns in the training set.

One of the ML methods we decided to apply during this initial exploration is Decision Trees. A decision tree is a type of supervised machine learning model that classifies data based on a series of binary decisions, with each level of the tree filtering out incorrect classes and coming closer to an accurate decision. This method is easy to visualize and serves as a great pilot method for a machine learning project, because it allows scientists to view the steps the model uses to classify data.

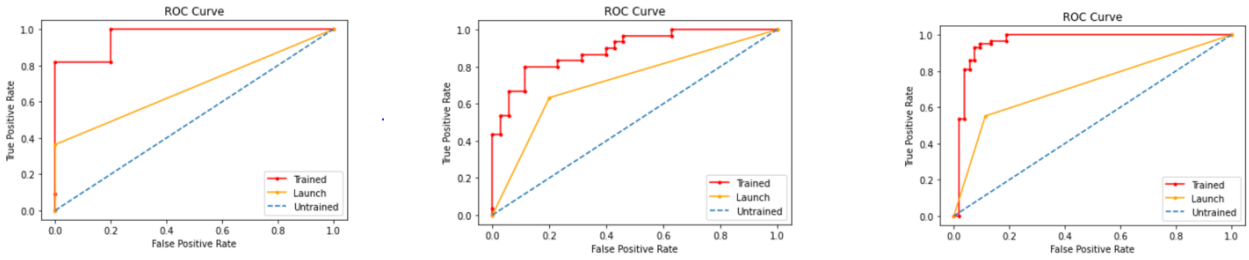
In our case, the data is each article, represented as a list of words, each with a TF-IDF value. At each level of the tree, a ‘splitting measure’ is used to decide which feature is to be used as the deciding factor to move to the next level down. These splitting measures can be measures such as Entropy, Gini Index or Information Gain, but these are all different methods of selecting what is basically the most important factor at each node. Moving down the tree leads to a reduction in the uncertainty of classification, which means that at the top of the tree, the initial split is made based on the word which is most important in the overall data set to be able to decide whether the article is innovative or not. In our case, this word was ‘Launch’ (**see Exhibit 4**). Therefore, we decided to use the presence of the word ‘launch’ as a naïve indicator for innovation, that our machine learning models could compete with.

Exhibit 4. Comparing ML with classification methods with naïve methods (e.g., use of word "launch" to identify innovation articles.



We use Receiver Operating Characteristic (ROC) plots to illustrate model performance. The ROC curve plots the performance of a model in terms of the error rate over increasing probability thresholds. The area under the curve (AUC) associated with a ROC curve is used to measure model accuracy. The straight diagonal line indicates a 50/50 guess and is used here as a baseline for performance. (**See Exhibit 5).**

Exhibit 5. Receiver Operating Characteristic (ROC) plots to illustrate model performance for three sectors studied



As we can see in the charts, the trained BERT methods (outermost red line on the ROC plot) perform better than a naïve word detection model (the middle yellow line), and significantly better compared to an untrained model (the diagonal blue line) and which indicates that BERT is able to pick up on nuances and context of text and recognize innovation.

Classification Methods for different economic sectors

Different economic sectors tend to talk about their products or industries differently. For example, the word 'launch' is commonly used in pharma ("Pfizer launched a new cancer drug for Leukemia"). In contrast, the word 'release' is often used in software, e.g., "Microsoft releases Flight Simulator 2020".

This variation and specificity of language mean that there is no one-size-fits-all approach to text modeling, especially when trying to detect subtle patterns like innovation. Therefore, we have applied different language models to other economic sectors to maximize our model performance (see **Exhibit 6).**

Exhibit 6. Classification Methods for different economic sectors to maximize model performance

Models (Pharma sector)	Accuracy
Pharma DistilBERT	71%
Pharma RoBERTa	72%
Pharma BERT	68%

Named-Entity Recognition (NER)

Once we can classify innovation vs. non-innovation articles, we can apply the model to all our articles. This approach gives us a set of innovation articles for industry each year.

This model needs to be revised to measure product innovation by companies because large companies with a high budget for Press releases tend to put out significantly more articles. To disambiguate and normalize results, we need to know which companies are mentioned in these articles and what products the articles describe. For this task, we utilize an approach called Named-Entity Recognition or NER (https://doi.org/10.48550/arXiv.2101.11420)

Named-entity recognition (NER), also known as (named) entity identification or entity chunking, and entity extraction) This method is a subtask of information extraction that seeks to locate and classify named entities mentioned in unstructured text into pre-defined categories such as person names, organizations, locations, etc.

Typically, NER uses eight categories—location, person, organization, date, time, percentage, monetary value, and "none-of-the-above." NER first finds named entities in sentences and declares the category of the entity. In the sentence:

> "Apple [Organization] CEO Tim Cook [Person] Introduces 2 New, Larger iPhones, Smart Watch at Cupertino [Location] Flint Center [Organization] Event."
> Note that "Apple" is recognized as an organization name instead of a fruit in its context.
> Bert-base-NER is a fine-tuned BERT model that is ready to use for Named Entity Recognition and achieves state-of-the-art performance for the NER task. It has been trained to recognize four types of entities: location (LOC), organizations (ORG), person (PER), and Miscellaneous (MISC).

Specifically, this Bert-based model was fine-tuned on the English version of the standard CoNLL-2003 Named Entity Recognition dataset.[2] (Sang et al. 2003). NER Identifies and categorizes entities into four categories based on the context in which the words are used and surrounding words in a sentence. It also tokenizes words. For example, Craftsbury becomes Crafts## and ##bury. It is always case sensitive and can be further fine-tuned by training on a corpus.

[2] https://paperswithcode.com/dataset/conll-2003

Training

The pretrained BERT-NER model is designed to identify specific types of entities: places, people, organizations, and various entities. The organization entity type is of interest to us, as company names get tagged with the NER-ORG tag, and we can assign a company name to each article. We use these NER-ORG tags to match against the human-labeled company names to measure how accurately the NER model has identified the correct company name.

Testing and Refinements

Compared to our human-labeled training set as ground truth, the results of implementing NER gave us promising results for extracting company names from articles (see **Exhibit 7**).

Exhibit 7. Named-Entity Recognition (NER) used to identify company names.

	Pharma	Food & Beverage	Software
Total Number of Labeled Company Names:	32	129	232
Exact Matches:	12	36	112
Fuzzy Matches:	14	73	80
Total Matches:	26	109	192
Total Accuracy (%):	**81%**	**84%**	**83%**

Since the company names in the article may be in different forms (Coke vs. Coca-Cola Inc.), we also use fuzzy matching to ensure that the company extracted by NER is the same as that identified by a human labeler (Arias 2019). We use similar fuzzy matching for the Question Answering method as well, to make sure we are matching product names correctly.

Using Fuzzy Matching for Company and Product accuracy testing

Fuzzy matching is a method to process word-based matching queries to find matching phrases from a set of articles, such as company names. Although not an exact match, fuzzy matching seeks to find a match above a threshold matching percentage specified by the program. The reason that fuzzy matching is important is that the same company or product may be labeled in subtly different ways or repeated in a piece of text such that the NER model might pick a different word than the human labeler. (For example, a human might label the company as 'Pepsi' vs. the NER model that might find the word 'Pepsi-Co' in the text and consider it as the most appropriate word to describe the organization). This disambiguation is also important to ensure we don't consider these two terms as two different companies.

Exhibit 8. Fuzzy Matching compared to Human-Labeled and Machine-Extracted Company Names

Innovators	Other_Companies	parsed_sentence	labels_for_sentences	Orgs from NER	Misc from NER	match ratios	highest match
23andMe	NaN	[gattaca, -, style, california, 23andme, new, ...	[B-MISC, B-MISC, B-MISC, B-MISC, B-ORG, B-LOC,...	[23andme, genepeeks, bbc]	[gattaca, -, style, california, -, based, gatt...	[(23andme, 100), (genepeeks, 25), (bbc, 0)]	(23andme, 100)
Innovus Pharmaceuticals	NaN	[bassam, damaj, innovus, pharmaceuticals, apea...	[B-PER, I-PER, B-ORG, I-ORG, B-MISC, B-MISC, B...	[innovus, pharmaceuticals]	[apeaz™, apeaz™, apeaz™]	[(innovus, 90), (pharmaceuticals, 90)]	(innovus, 90)

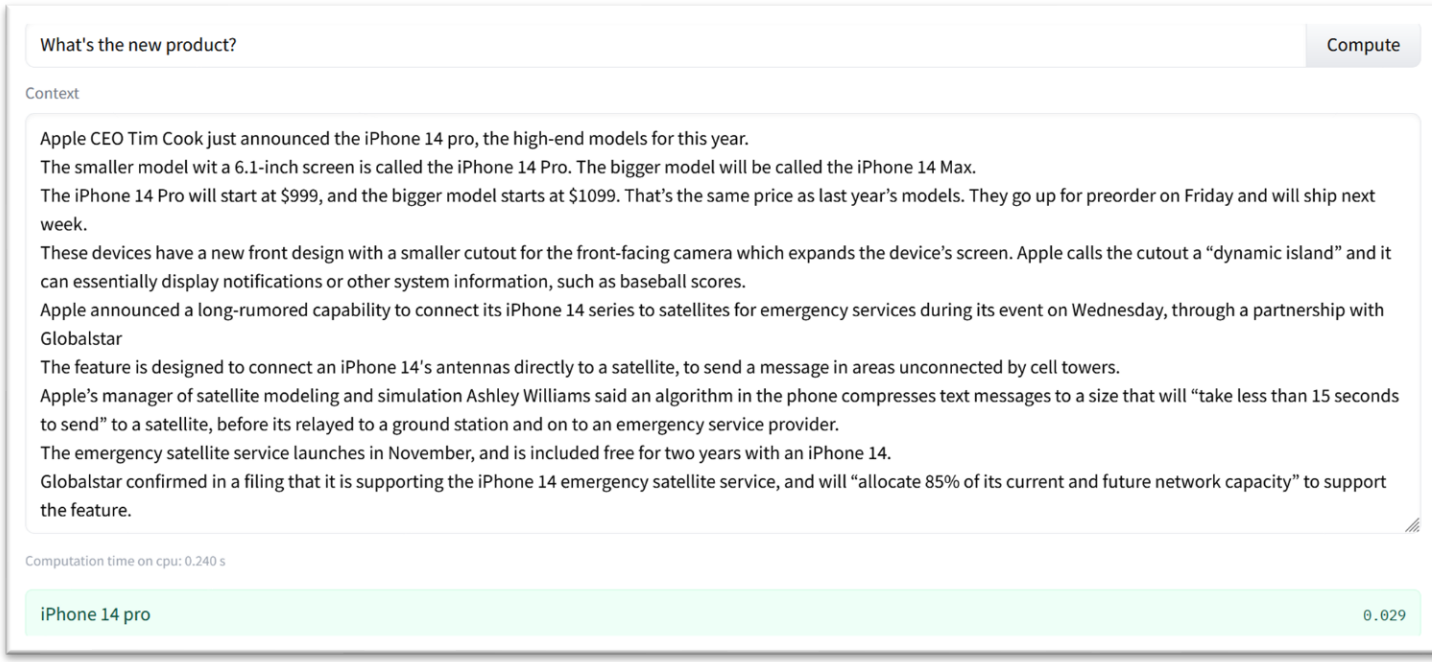
In the above examples, we use fuzzy matching to compare the human-labeled and the machine-extracted company names. We can see in the second example that Innovus Pharmaceuticals is matched with the word 'Innovus' with a 90% match confidence (see Exhibit 8).

Question Answering (QA)

BERT uses token embedding similarities to retrieve answers to a given question from a reference corpus. This method works similarly to a reading comprehension question in English, where the answering system tries to find the correct answer from the provided context**.**

Exhibit 9 illustrates the question asked at the top, the context text that the model searches for the answer, and the answer highlighted in green at the bottom. This approach relies on sentence similarity, where the sentence-level embeddings of a question have a high similarity score to the sentence-level embeddings of a sentence that contains the answer to the question. The questions are used to extract specific information from a given text corpus.

Exhibit 9. Question-Answering (QA) example to detect company name and product.



Pre-trained model -- Hugging Face

The model we chose for question answering was the current state of the art for the QA task. We used the deepset-roberta-base for QA, fine-tuned using the SQuAD2.0 dataset. It's been trained on question-answer pairs, including unanswerable questions, for the task of Question Answering (Hugging Face 2022, Stanford NLP Group 2022).

QA Implementation and Testing

Using a single question gives us moderate accuracy. To improve accuracy, we can use multiple questions designed to converge on the same answer. For example: "What is the name of the company that produced the innovative product?" can be rephrased as "What is the name of the innovative company?" or "Who is the innovator?" which are all subtly different but should converge to the same answer when given the same article as context. The answers to each of these questions are fed to a voting system; if most of the answers are the same, that is considered the correct answer. If all three have different answers, the question with the highest individual performance record is considered the right answer (see **Exhibit 10).**

Exhibit 10. Using multiple questions to improve the QA process

	title	Lead Paragraph_512	names_company_about_list	Innovators	What's the new product?	What's the new product? **probability**	what's the company name?	what's the company name? **probability**	what's the new drug?	what's the new drug? **probability**	...	Which company announced the product? **probability**	When will the company announce the product?	When will the company announce the product? **probability**
0	MEDTECH : MEDTECH ANNOUNCES NEW SALES OF THEIR O...	The ROSA™_Brain system at the Yale Comprehensi...	[['Medtech SAS']]	NaN	ROSA™ Spine	0.978814	Medtech	0.798178	ROSA™ Spine	0.673567	...	0.737372	January 2016	0.710806
1	CUBA SEEKS MALAYSIAN COLLABORATION VIA PHARMAC...	She said Cuba viewed Malaysia as a significant...	[]	NaN	lung cancer vaccine	0.580466	Cimavax	0.732677	lung cancer	0.334009	...	0.737267	next year	0.359116
2	Stents appear to increase stroke patients' rec...	Currently, standard stroke care in the United ...	[]	Medtronic PLC	stent-based clot removal	0.293876	Medtronic	0.848025	tPA	0.865578	...	0.975851	Bloomberg Businessweek	0.662318
3	PSIVIDA CORP. pSivida to Present At Two Invest...	pSivida will also present at the Stifel Nicola...	[['EyePoint Pharmaceuticals Inc'], ['Direct Ma...	EyePoint Pharmaceuticals Inc	injectable, sustained release micro-insert ILU...	0.298959	pSivida Corp	0.317354	Latanoprost	0.962495	...	0.439559	Thursday, September 12	0.404281
4	Strides Shasun receives US FDA approval	Strides is launching the product immediately. ...	[['U.S. Food and Drug Administration'], ['Stri...	Strides Pharma Science Ltd	Tenofovir Disoproxil Fumarate	0.928353	Strides Pharma Inc	0.451422	Tenofovir Disoproxil Fumarate	0.881826	...	0.946747	immediately	0.813833
5	Gilead Sciences Inc Files Patent Application f...	The abstract of the patent published by the Co...	[['Gilead Sciences Inc']]	Gilead Sciences Inc	experimental drug candidate	0.425972	Gilead Sciences, Inc.	0.505565	HIV/AIDS	0.851420	...	0.527109	July 31, 2015	0.783506

Text Pre-processing

To clean the text for better analysis, we utilized several different filters to uniformly treat the text data based on what we saw as commonly irrelevant for identifying a company. Intuitively, we accomplish this by applying basic word cleaning, then progressively more aggressive filters. Then we retrieve the shortest string with a length **greater than zero among all the filter steps. For example, consider the sequences shown in Exhibit** 11.

Exhibit 11. Word cleaning to improve accuracy of identifying company name

Example	Lower case company name	(A)Alphanumeric	(B) Remove common abbreviations	(C) Filter Locations	Shortest Result
1	†wal-mart stores, inc.†	walmart stores inc	walmart stores	walmart stores	walmart
2	-l'oreal usa products inc	loreal usa products inc	loreal products	loreal products	loreal products
3	amsino healthcare (usa) inc	amsino healthcare usa inc	amsino healthcare	amsino healthcare	amsino healthcare
4	shenzhen lantern science co. ltd.	shenzhen lantern science co ltd	shenzhen lantern science	lantern science	lantern science

Here, we see that the alphanumeric character-only filter (A) in examples 1 and 2 effectively removes extracted character artifacts but could add noise to identifying company names. We remove a list of common-company-abbreviations in filter (B) that natural language processors have difficulty addressing. For example, things like tech or "us" versus "usa" can be effectively filtered with a dictionary. Then for the location filter (C), we observe many samples where the company's location is added but bears no weight on the name of the company itself. For examples 2, 3, and 4, we showcase how filtering out country and city names from the extracted string can lead us to a more condensed list of company

names. To see the implementation of the company name cleaning filters and more samples, please visit our GitHub repository.

Does the article title include all the innovation information we need?
BERT has a limitation in the amount of text it can process at a time, which is 512 tokens. This limitation means that using traditional BERT models limits us from analyzing the title and lead paragraph of the text, not the whole text. New implementations in 2022 have overcome this limitation, but this poses the question, does the article title have sufficient information to classify innovation and extract company and product names? Most news articles on the internet are front-loaded with relevant information, i.e., they place distinguishing information at the beginning of headings, paragraphs, lists, etc.) (Dwyer, C., Frederico, S. 2015. American Press Institute 2014, WAI 2000)

To test this, we have compared accuracy using only title text vs. title and lead paragraph text. initially tested this idea based on an initial sample of 32 true positives. Using titles only, the accuracy was 73%. Using titles and lead paragraphs, the accuracy was 87%.

This would suggest that it could be a good idea to start by analyzing text titles to get an idea of patterns before acquiring the full article text to get a company name and product names. By using metadata to choose better training sets, the number of True Positives Innovation examples increased to 277 true positives. Training on that, we obtained the following results shown in **Exhibit 12**. We used 3 different language models, DistilBERT, BERT and RoBERTa. DistilBERT is the smallest and least complex model, and consequently has the fastest execution time, RoBERTa is the largest and most complex of the three, with the longest execution time, while BERT is the median in terms of complexity, size and runtime. DistilBERT is able to predict innovation in our sample 72% of the time using only the title, but 71% of the time using title and lead paragraph. This seems to indicate that the less complex language model is able to pick up on the innovation signal in title text, but the addition of the lead paragraph introduces noise that makes the classification slightly more challenging. BERT also repeats this trend where Title and lead paragraph has a lower classification accuracy than just Title. RoBERTa, with a more complex vocabulary, is able to process the lead paragraph data better, and obtain a higher accuracy than just the title. RoBERTa was previously found in exhibit 6 to be the best candidate model for Pharmaceutical articles.

Exhibit 12: Classification Model performance.
Based on Labeled Pharmaceutical articles with 277 true positives for innovation +

DistilBERT	
Title only	72%
Title and lead paragraph	71%
BERT	
Title only	70%
Title and lead paragraph	68%
Roberta	
Title only	63%
Title and lead paragraph	72%

These results indicates that for certain language models, it may be sufficient to use article titles to classify if the article is innovative or not. This may reflect the way articles are written on the internet to cater to readers who do not have the time to read through a full article (American Press Institute 2014).

Comparison of NER and QA Performance

Using a data set of 600 articles related to the software industry, we compared the innovating companies extracted by NER and QA. Extracted company names may come with prefixes or suffixes, so we applied the fuzzy matching method and set the threshold for a "match" to be at least 80% weighted ratio (WR), calculated through the Levenshtein distance between two names. The results show that 36% of QA results matched the NER results, and another 47% passed the fuzzy match, yielding a total matching rate of 83%.

Since the NER algorithm outputs a list of possible entity names, each with a score, we adopted a strict matching test in addition to the standard fuzzy matching. When comparing only the highest-score name extracted by NER to the QA result, the total matching rate dropped to 50%. This finding constitutes the lower bound consistency between the two algorithms.

When comparing Company Names extracted by NER, we tried comparing between the company extracted by NER to the Company labeled by Human labelers. Recall that NER can be used to extract specific categories of data, such as organizations. We see that the match between NER extracted values and manually labeled company names is higher when using Title and Lead paragraph, but also quite high for Lead paragraph only, which reflects how information in articles is often front-loaded in the lead paragraph (Dwyer, C., Frederico, S. 2015. American Press Institute 2014, WAI 2000). See **Exhibit 13.**

In **Exhibit 14** we have the Question-Answering results. This method is able to extract product name as well as company name, and through the use of specific question phrasing, we are able to get more accurate results for both Company and Product name, though this process is more time consuming (around 10 times the execution time as NER)

Exhibit 13. NER Performance: Company Name Matching Rates for Software Industry

Company Name Matching Rates	**Manually Labeled Company Names**
Title only	62.5%
Lead paragraph only	87.5%
Title and lead paragraph	90.6%

Note: Subset of Article Pre-labeled Column for Comparison. Processed Hand-Labeled Innovator Companies from DNA

Exhibit 14. QA Performance: Company and Product Name Matching

Company and Product Name Matching	**Company**	**Product**
Individual answers (title + paragraph)	78%	81%
Combined answers (title + paragraph)	97%	91%
Combined answers (title only)	94%	91%

Conclusion

In this research, we tested the use of BERT and related products and classifiers to identify product innovations and innovative companies in news articles. We have three main conclusions.

- It is possible to use modern language models, e.g., BERT, to detect subtle patterns in text, such as innovation.

- Using modern language models significantly outperforms naïve approaches such as keyword search (on words such as ‘launch’ or ‘new’) when applied to the problem of innovation detection.
- We can use Named Entity Resolution (NER) and Question-Answering (QA) methods to extract specific information from the article text, such as company name or product name, in a consistent and repeatable way.
- Much information can be obtained using the article title rather than the full text. This implies that when applying these techniques to a new task or data domain, a first-pass analysis could be made using article titles that are easier to obtain and faster to process. If the first-pass analysis is promising, one could use the abstract or full text for a more in-depth analysis.
- In our current approach, a sector specific approach is used both for data cleaning and for building a classification model that can identify innovation. This is due to the fact that different sectors have different language patterns in their articles.
- While language models perform significantly better than naïve approaches, the use of complex language models does not necessarily result in better performance in all sectors. The increased computational time and cost may not justify the marginal increase in accuracy.
- Models trained in one sector do not necessarily work in other sectors. A ‘generalized’ innovation model is theoretically possible but would need a huge amount of labeled training data to actually develop which is practically challenging. This may change as language models get increasingly sophisticated.

Next Steps

One of this project's challenges is that news articles are inconsistent across industries and news sources. Finding a consistent source of company information, such as SEC filings, could be a promising data source to apply BERT methods. The challenge in using SEC filings is retrieving and parsing the information from the EDGAR source. Additionally, SEC filings may be too long for BERT to process. However, we could overcome both limitations with new software packages for data retrieval and more recent versions of BERT that use 'Longformers' (Transformers that can be applied to longer data streams).

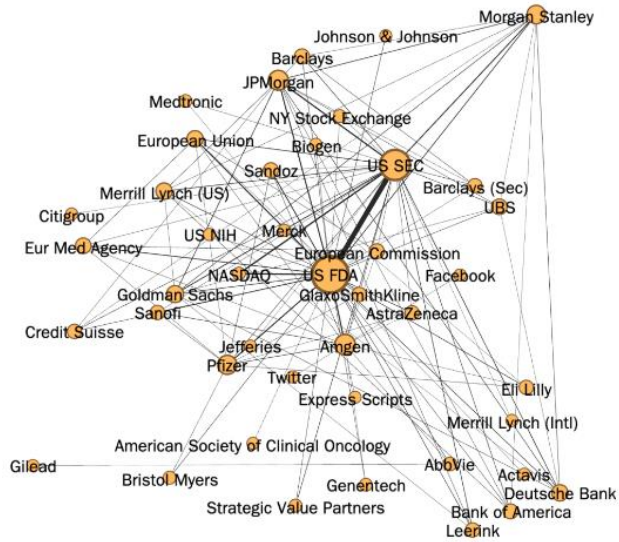
We could use language models to conduct document similarity tests to find relevant documents from a large corpus. For example, suppose we want to find documents about the topic ‘COVID-19. In that case, we could find at least one article or abstract related to the topic and then use BERT or BERT-like language models to find related articles. This helps to find the most relevant articles from the corpus, which will make the process of human labeling more efficient and can be used to train classification models that can decide whether an article is related to a specific topic.

We could also test new approaches to assessing and benchmarking the innovation measurements. We could build an innovation index for a given company for a given year, compare that company's performance to the top 100 companies in the same sector, and see if the companies we detect as innovative have a better performance than the average. This approach could reinforce the validity of our findings.

An ongoing challenge is the question of company matching. A company often uses different names across different data sources (e.g., "Eli Lilly & Co." in the FDA set, as opposed to "Eli Lilly" in the NDC set). Therefore, it can be challenging to identify the same company across datasets. Building a matching dictionary that could allow us to disambiguate company names would help mitigate this issue.

Network Analysis to explore links across products or companies to explore questions about interactions between organizations (such as, *are competitors in a sector part of the same networks, or are they isolated from each other?*). A sample network for a subset of companies was generated as part of the analysis, which looks at innovative companies in the pharmaceutical sector and the companies co-mentioned in the same article. This network analysis illustrates the relationships across financial and pharmaceutical companies, regulatory agencies, and other organizations (see **Exhibit 15**).

Exhibit 15. Network Analysis illustrates the Relationships across Financial and Pharmaceutical Companies, Regulatory Agencies, and Other Organizations



We have built tools for accessing news articles and obtaining SEC filings in a convenient, open-source format. These tools can be made available for researchers who want to expand on this work and apply it to different sectors. Once tested, this would provide a free alternative to commercial datasets.

- SEC Filing scraper (https://github.com/uva-bi-sdad/sec)
- RSS Scraper (see appendix 1) (https://github.com/uva-bi-sdad/rss-scraper)

Appendix 1: An Open-Source method to obtain news article text for specific keywords and time ranges using publicly accessible RSS Feeds.

The Product Innovation project is a proof-of-concept toolkit that aims to track innovation activities sustainably using opportunity data (Keller et al. 2020). Using open-source modules and browser automation, the toolkit accelerates Really Simple Syndication (RSS) queries and news source text extraction. We then applied natural language processing (NLP) to analyze the collected texts to detect business, product, and innovation status.

Our project aimed to explore the feasibility of complementing the Annual Business Survey (ABS) with alternative data sources. The National Center for Science and Engineering Statistics (NCSES), part of the National Science Foundation (NSF), conducts the Annual Business Survey to collect data on R&D, innovation, technology, intellectual property, and business owner characteristics.

While ABS measures innovation incidence, i.e., the number of innovating firms, we aim to test the feasibility of developing methods using non-traditional data to obtain richer and complementary innovation measures [1. As] part of this, we investigated the use of opportunity data on the web. While accessing websites individually as humans is easy, we found it non-trivial to automate news-text extraction in a free and open-source way. Consequently, we contribute this summer by creating an example framework for researchers to extract news text.

Product Innovation: Defined

To decide whether a news article is innovation related, we must first define what we mean by product innovation. For this, we use the definition contained within the Oslo Manual. First published in 1992, the Oslo Manual is the international reference guide for collecting and using data on innovation (OECD/Eurostat 2018)

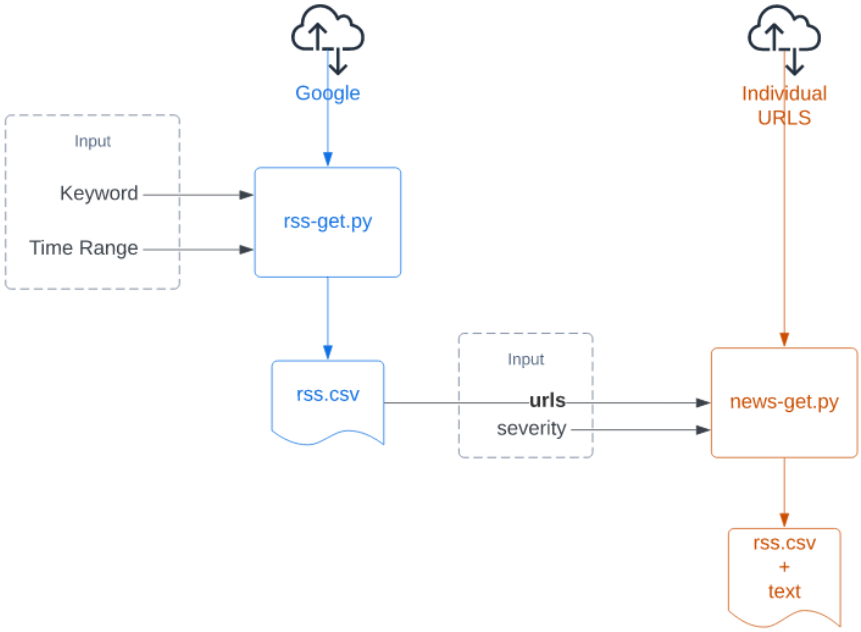
Innovation is defined in the Oslo Manual as: "New or significantly improved goods or services that are available to the general public of users." For example, even if a company recreates a drug once its patent protection expires, it is still considered an innovation because it is new to the company and a new product to the market, which can advance the economy. Conversely, completing a patent or FDA approval process is not considered an innovation if the creation does not go to market.

The scope of our work would be limited to just US-based innovations. Our first task was to create a python tool that, given the input of a keyword, extracts information about its 1) title, 2) URL, 3) time of publication, and 4) the first paragraph.

Methods

We created two modules, the RSS-get module, and the **news-get** module. The **rss-get** takes in keywords and returns URLs, and the **news-get** takes in URLs and returns source text. Below, we document how we arrived at our comparison of search engines, our comparison of source text extractors, and the concept of *severity* which we use to classify the difficulty of websites from being scraped (see Exhibit 16).

Exhibit 16. Two modules identify and retrieve innovation articles: the RSS-get module takes in keywords and returns URLs; the news-get takes in URLs and returns source text.



Severity Levels

As we collected source text, we realized that news sources have a varying degree of "friendliness" to being scraped. In other words, some websites intentionally resist source text from being extracted. We introduced the concept of severity to our system so that we can navigate spending more computational power to access these more difficult websites.

[Reference website https://uva-bi-sdad.github.io/dspg22/product-innovation/ for further details on websites, sources, etc.]

About the University of Virginia's Social and Decision Analytics Division

The Social and Decision Analytics Division (SDAD) is a leading Division in the Biocomplexity Institute at the University of Virginia. The Biocomplexity Institute is at the forefront of scientific evolution, applying a deeply contextual approach to answering some of the most pressing challenges to human health and well-being within our changing environment. SDAD was created in the fall of 2013 to extend the Biocomplexity Institute's capabilities in social informatics, policy analytics, and program evaluation. The researchers at SDAD form a multidisciplinary team with expertise in statistics, policy and program evaluation, economics, political science, psychology, computational social science, data governance, and information architecture. SDAD's mission is to embrace today's data revolution, developing evidence-based research and quantitative methods to inform policy decision-making and evaluation.

Acknowledgments

This research was conducted under US National Science Foundation, National Center for Science and Engineering Statistics contract #49100420C0015.

References

Amazon mTurk (2022) Pricing. https://requester.mturk.com/pricing

American Press Institute. (2014, Mar). Chapter 2: How Americans get their news. https://www.americanpressinstitute.org/publications/reports/survey-research/how-americans-get-news/

Arias, F., (2019, Feb) "Fuzzy String Matching in Python," DataCamp, https://www.datacamp.com/community/tutorials/fuzzy-string-python

Bellstam, G., Bhagat, S., & Cookson, J. A. (2021). A text-based analysis of corporate innovation. *Management Science*, *67*(7), 4004-4031. https://pubsonline.informs.org/doi/abs/10.1287/mnsc.2020.3682

Beltagy, I., Lo, K., Cohan, A. (2019) "SCIBERT: A Pre-trained Language Model for Scientific Text," arXiv, 10-Sep-2019, https://arxiv.org/pdf/1903.10676.pdf

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. https://arxiv.org/abs/1810.04805

Dwivedi, P. "Testing BERT-based Question Answering on Coronavirus articles," Medium, 06-Apr-2020. https://towardsdatascience.com/testing-bert-based-question-answering-on-coronavirus-articles-13623637a4ff.

Dwyer, C., Frederico, S. (2015, Oct). Keep Readers Engaged. Tips for writing better headlines. NPR Training/Sources. https://training.npr.org/2015/10/25/the-checklist-for-writing-good-headlines/

Héroux-Vaillancourt, M., Beaudry, C., & Rietsch, C. (2020). Using web content analysis to create innovation indicators—What do we really measure? *Quantitative Science Studies*, *1*(4), 1601-1637. https://direct.mit.edu/qss/article/1/4/1601/96113/Using-web-content-analysis-to-create-innovation

FDA (2022a) "Drugs@FDA Data Files," U.S. Food and Drug Administration, https://www.fda.gov/drugs/drug-approvals-and-databases/drugsfda-data-files

FDA (2022b) "National Drug Code Directory," U.S. Food and Drug Administration, https://www.fda.gov/drugs/drug-approvals-and-databases/national-drug-code-directory

Hugging Face (2022) "robert-based-squad2." https://huggingface.co/deepset/roberta-base-squad2

Keller, S. A., Shipp, S. S., Schroeder, A. D., & Korkmaz, G. (2020). Doing Data Science: A Framework and Case Study. Harvard Data Science Review, 2(1). https://doi.org/10.1162/99608f92.2d83f7f5Kinne, J., & Lenz, D. (2021). Predicting innovative firms using web mining and deep learning. *PloS one*, *16*(4), e0249071. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8016297/

Kaput, M. (2021) What Is Google BERT? Experts Explain. https://www.marketingaiinstitute.com/blog/bert-google

Lee, J., Yoon, W., Kim, D., Kim, S., Kim, So, C., Kang, J. (2019, Sep) "BioBERT: a pre-trained biomedical language representation model for biomedical text mining," arXiv, 1, https://arxiv.org/ftp/arxiv/papers/1901/1901.08746.pdf

NSF NCSES. (2022). Product Innovating Companies. 2017-2019, Table 24. nsf22344-tab024. US National Science Foundation (NSF), National Center for Science and Engineering Statistics. https://ncses.nsf.gov/pubs/nsf22344

OECD/Eurostat (2018), Oslo Manual 2018: Guidelines for Collecting, Reporting and Using Data on Innovation, 4th Edition, The Measurement of Scientific, Technological and Innovation Activities, OECD Publishing, Paris/Eurostat, Luxembourg. https://doi.org/10.1787/9789264304604-en

Ratcliff, N. & Kattampallil, N. (2022). A labeling methodology for identifying business product innovation in pharmaceutical articles. Proceedings of the Biocomplexity Institute, Technical Report. TR# 2022-013 University of Virginia. https://doi.org/10.18130/eps9-t189

Arya Roy, Recent Trends in Named Entity Recognition (NER), Carnegie Mellon University https://doi.org/10.48550/arXiv.2101.11420

Sang, E. F., & De Meulder, F. (2003). Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. arXiv preprint cs/0306050 and https://paperswithcode.com/dataset/conll-2003

Stanford NLP Group. (2022) “SQuAD 2.0,” SQuAD - the Stanford Question Answering Dataset. https://rajpurkar.github.io/SQuAD-explorer/explore/1.1/dev/ .

Sterbak, T. (2020, April). “Named-entity recognition with Bert,” Depends on the definition, https://www.depends-on-the-definition.com/named-entity-recognition-with-bert/

Tjongkimsang, E., Ddemeulder, F. (2005). Named-Entity Recognition (II). https://www.clips.uantwerpen.be/conll2003/ner/

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, *30.* https://arxiv.org/abs/1706.03762v5.

WAI (2000). Example for Checkpoint: 13.8 - Place distinguishing information at the beginning of headings, paragraphs, lists, etc. Web Accessibility Initiative. https://www.w3.org/WAI/wcag-curric/sam110-0.htm