Reporte de proyecto ESG: Análisis de texto y tópicos.

En este reporte se detallan los aprendizajes, dificultades encontradas, problemas, soluciones y alternativas al replicar el articulo en estudio, utilizando datos de empresas chilenas.

I. Extracción de texto.

En primer lugar se necesita extraer los datos de su fuente y limpiarlos de manera que sea sencillo procesarlos.

Para esto inicialmente utilizamos pyPDF2, una librería típicamente recomendada para esta tarea, inclusive es la más herramienta más popular en los resultados de la búsqueda por internet a la consulta “text extraction from pdf in Python”.

Sin embargo, en los resultados posteriores sorprendía la baja calidad de los resultados, y analizando etapa a etapa se llegó a que el problema se encontraba en la extracción de texto, en particular que no se estaba extrayendo en su totalidad, más bien solo una pequeña parte. La razón de esto es desconocida, ya que el uso de este paquete para los autores del articulo original no tuvo inconvenientes, suponemos que tiene que ver con el formato que utilizaron las empresas para sus reportes, ya que el parseo de pdf es bastante delicado y complejo.

Como utilizar el paquete pyPDF2 ya no era opción, encontramos la siguiente mejor alternativa: tika.

Tika es un paquete de parsing de texto en archivos pdf, promete realizar lo mismo que pyPDF2. En la practica encontramos que los resultados obtenidos son mucho mejores con este paquete, se extrae el texto casi en su totalidad. Aunque no todo fue buenas noticias, si bien ahora se extrae de mejor manera el texto del reporte, tika está repitiendo parte del contenido extraído múltiples veces (inicialmente sin que se tuviera conocimiento de esto). Por lo que los pasos siguientes, como la limpieza, lematización y entrenamiento de modelos tardó muchas horas en completar, cuando al no haber texto duplicado tan solo se tarda unos minutos el proceso entero. Se destaca como aprendizaje futuro el realizar una inspección más detallada de los datos extraídos y con los que se va a trabajar, para economizar en tiempo de cómputo y productividad.

Lo positivo que se rescata de este error, además del aprendizaje futuro para no cometerlo nuevamente, es que mientras se consideraba que la cantidad de texto que se estaba procesando era correcta y no habían duplicados, se optó por experimentar con procesar los datos en paralelo. En particular realizar la lematización del texto por trozos, mediante múltiples procesos de cpu. Se darán más detalles más adelante, luego de presentar el tema de lematización.

II. Procesamiento y limpieza de texto.

Para realizar la limpieza de texto se implementó una función de estilo .apply de pandas, por filas, pues de esta manera se hace en el articulo de los autores originales. Sin embargo, dados los resultados posteriores, en particular el tiempo que se tardaban en obtener, se llegó a la conclusión de que esta no era la forma óptima de hacer el procesamiento, y así llegamos a los métodos .str de pandas.

Estos métodos reciben el nombre de funciones vectorizadas, y en comparación a una función regular están limitados en lo que se puede realizar con ellas y la manera en que se deben programar. En vez de aplicarse por filas, se aplican sobre una columna, y están optimizadas para ser cientos de veces más rápidas que procesar dato a dato mediante una aplicación .apply por filas (fuente)

III. Lematización

Teniendo el texto preparado solo queda pasarlo a un modelo NLP para lematizarlo. Para ello inicialmente usamos spacy, una herramienta popular para este trabajo que también es usada y recomendada por los autores del artículo.

Lamentablemente no tuvimos buenos resultados con este modelo, spacy no era capaz de reconocer correctamente el POS de las palabras en muchas ocasiones, resultando en lematizaciones como AGUA->AGUAR, lo que comprometía por completo nuestros resultados.

Nuestra hipótesis es que los autores no tuvieron problemas ya que ellos trabajan con textos en inglés, el idioma predeterminado para el entrenamiento de modelos grandes y robustos. Mientras

que en español es difícil encontrar modelos entrenados con grandes corpus de datos, ya que hay un público mucho menor que le encontraría utilidad, por lo que no se le da mucha importancia.

Obligados a encontrar una mejor manera de lematizar nuestro texto fue que encontramos stanza, un paquete de Python con modelos NLP entrenados por la universidad de Stanford. Este paquete tuvo muchas características positivas, la manera de utilizar su API es casi idéntica a spacy, como si siguiera un estándar. Adicionalmente, los resultados de la lematización eran claramente mejores, mediante observación no fuimos capaces de encontrar algún error entre oraciones.

Desafortunadamente un mejor modelo también implica un modelo más grande, y más tiempo de cómputo. Al cambiarnos a este paquete los tiempos de lematización aumentaron drásticamente, al punto de que procesar nuestro corpus pasó de tardar ~30 minutos a 17 horas.

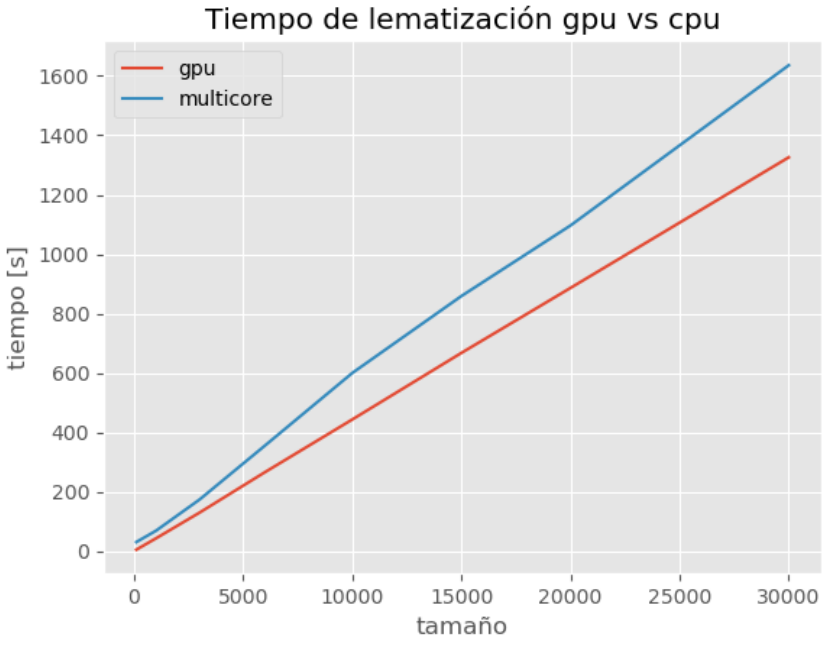
Como este trabajo es iterativo y es necesario volver a pasos anteriores del proyecto y ejecutar de nuevo el procesamiento, estos tiempos no iban a ser compatibles con lo que estamos realizando.

De esta manera llegamos a la idea de paralelizar la lematización que se realiza sobre el dataframe de datos, dividiéndolo en tantos trozos como núcleos de cpu se disponga y que cada uno procese su trozo de manera independiente, para finalmente juntarlos todos y obtener el resultado final.

Casi de manera simultanea nos percatamos de que stanza entrega la opción de procesar mediante gpu, lo que indica que es bastante probable que el modelo esté basado en una red neuronal. Como es el caso con las computaciones en gpu, inmediatamente asumimos que utilizarla mejoraría el tiempo de cómputo en gran medida.

Ahora nos encontramos con dos opciones, procesar utilizando una gpu, limitada a un solo proceso (pues no se puede paralelizar el uso de gpus, ya que es solo una), lo que implica lematizar una oración a la vez. O realizar la lematización mediante múltiples procesos de cpu, limitados a que estos no pueden hacer uso del modelo en gpu, por lo que cada proceso es más lento que la opción anterior.

Para encontrar la mejor alternativa realizamos un experimento, donde comparamos los tiempos de procesamiento de ambas opciones para distintos tamaños de dataset, y ver como escalan a medida que aumentan los datos a procesar.



Ambas muestran un comportamiento lineal a medida que se aumenta el tamaño de entrada, sin embargo, el uso de gpu es mucho más estable en términos de tiempo, y a su vez más rápido, que es lo que importa.

Además, se observa que mientras más grande el tamaño del dataset también la diferencia de tiempos entre el uso de gpu y múltiples procesos aumenta cada vez más, por lo que utilizar gpu se vuelve más tiempo eficiente entre más grande sea el set de datos.

Estos resultados se obtuvieron con un cpu Intel i7 4770, de 4 cores y 8 threads y una gpu nvidia GTX 1060 6gb. Se hipotetiza que una cpu más moderna, de 8 o 16 núcleos podría alcanzar un rendimiento similar e incluso superior al de la gpu que utilizamos. Pero también se podría utilizar una gpu más moderna, sobre todo con la salida de la última generación de RTX 3000, lo que mostraría mejorías en tiempo de gran magnitud.

IV. Análisis de texto

A partir de este punto se tiene una tabla de oraciones y oraciones lematizadas identificadas por compañía. Dependiendo del origen del texto que se esté utilizando puede que pasos adicionales de limpieza sean necesarios en este paso. En nuestro caso tuvimos que filtrar oraciones o

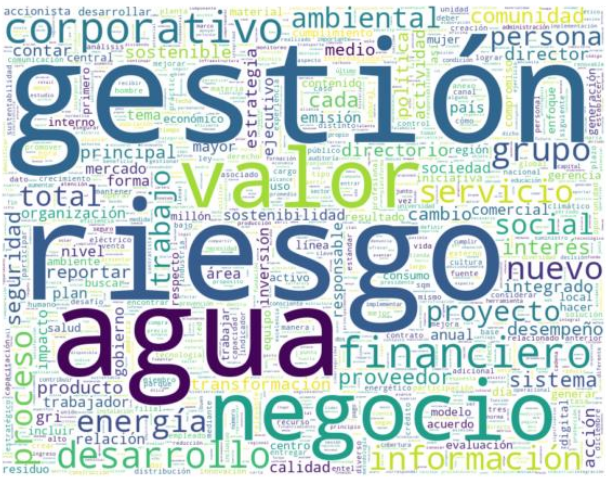
lematizaciones con muy baja cantidad de palabras y oraciones con errores de parseo (en muy pocas ocasiones se parseaba texto de la forma “s u s t e n t a b i l i d a d” cuando el resultado esperado es “sustentabilidad”. Esto hubiera generado que se considerara cada letra como una palabra, lo cual es errado.

La primera estadística por considerar es la incidencia de cada empresa dentro del corpus de texto que tenemos, ya que cantidades dispares en la cantidad de oraciones por empresa pueden indicar un sesgo hacia los temas mencionados por aquellas de mayor frecuencia.

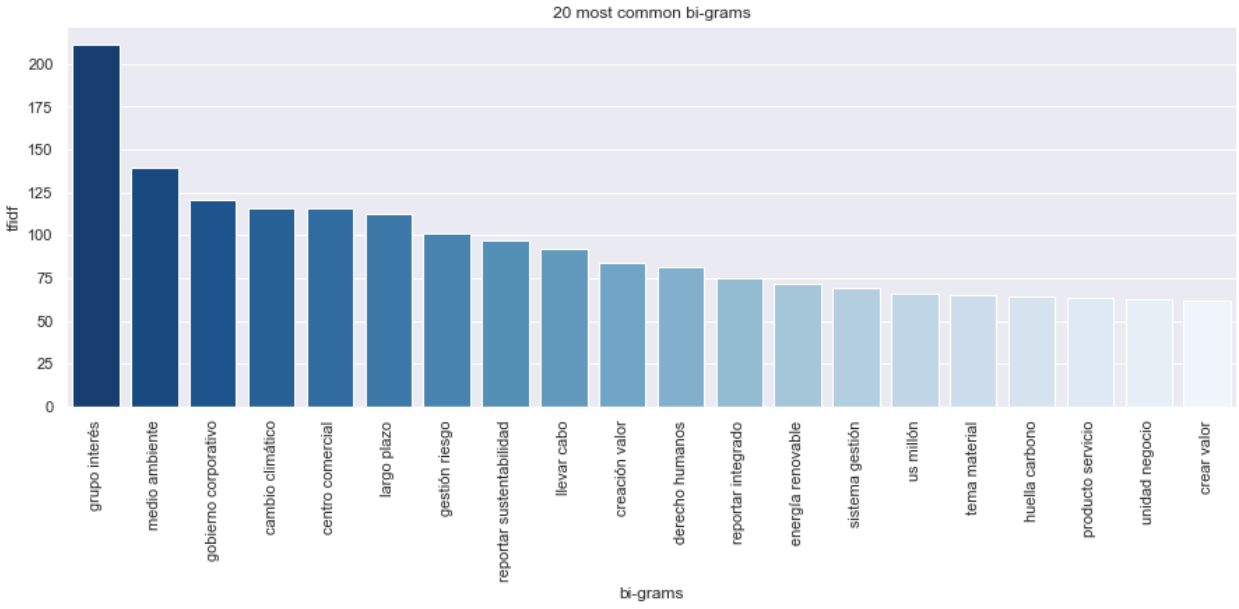
Compañía	Oraciones
endesa am	2201
aesgener	1838
santander	1836
sqm	1829
itaucorp	1543
aguas andinas	1380
colbun	1374
endesa-cl	1350
cencosud	1311
andina embotelladora	1247
entel	1079
falabella	1007
latam	929
cmpc	904
salfacorp	880
concha toro	730
sonda	718
bci	711
parauco	620
ecl	506
grupo security	405
ccu	217
cap	195
ilc	176
ripley	160

Luego interesa observar los unigramas y bigramas más comunes en todo el corpus de texto. Para ello realizamos una nube de palabras considerando solo unigramas y una considerando solo bigramas para visualizarlos.

Para forzar el paquete wordcloud a considerar exclusivamente bigramas se concatena todo el texto en pares de tokens, uniéndolos mediante guiones.



Adicionalmente mostramos la frecuencia de los bigramas más comunes para tener una idea de cuáles son los términos más utilizados de manera jerárquica.



Como ya se mencionó, este trabajo es iterativo, y una de las razones es que en este paso salen a luz muchas stopwords que pudieron pasar por alto o no se estaban considerando. Por lo que el

análisis de estas gráficas tiene utilidad tanto el desarrollo o implementación como en el análisis del producto final.

V. Topic Modelling

Lo siguiente que queremos realizar es poder agrupar cada oración de nuestro dataset dentro de una categoría, o tópico. Esto desde la hipótesis de que las empresas al reportar sobre sustentabilidad abarcan temas similares. Si esto se cumple se tendrán varios usos para esta categorización, esto se verá más adelante.

Para poder discernir nuestras oraciones entrenamos un modelo LDA utilizando scikit-learn. Sin embargo, previo al entrenamiento es necesario encontrar la cantidad optima de tópicos y parámetros de aprendizaje que optimizan los resultados, en particular las métricas de perplexity y loglikelihood, indicando que es el mejor modelo ajustado a nuestros datos.

Para encontrar estos parámetros utilizamos el algoritmo de búsqueda GridSearch, que en esencia entrena un modelo por cada combinación de parámetros a evaluar, y entrega la mejor de ellas. Los parámetros entregados fueron cantidad de tópicos y learning decay.

Hay que tener precaución al utilizar este algoritmo de búsqueda, ya que, si entrenar un modelo puede tomar bastante tiempo, este algoritmo multiplica ese tiempo por la cantidad de combinaciones a explorar. Por lo que es esencial acotar el espacio de búsqueda a valores que se tiene conocimiento están cerca del óptimo. En nuestro caso consideramos que la cantidad de tópicos debía estar por sobre 5 pero bajo 10, valores inferiores o superiores no harían mucho sentido, por lo que no vale la pena explorarlos y perder tiempo.

Finalmente obtuvimos que el mejor modelo ajustado a nuestros datos encontró 7 componentes (tópicos) con un learning decay de 0.7.

La siguiente tarea es identificar cada uno de los tópicos, y nombrarlos. Para ello examinamos las principales palabras de cada uno “keywords” para inferir de que tema se trata en cada caso.

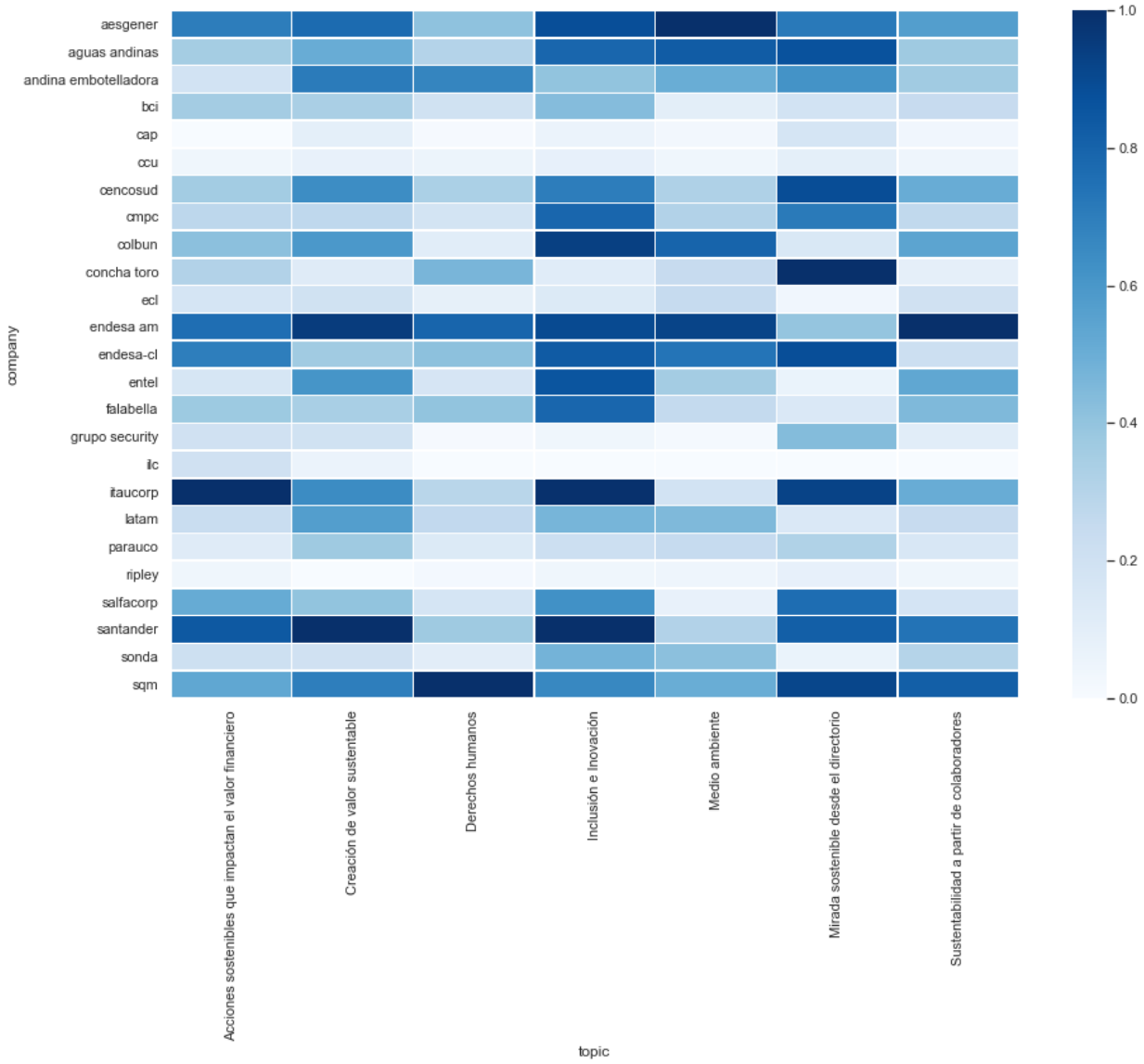
Tópico	Keywords
Topic #1:	director, ejecutivo, plazo, directorio, adicional, accionista, largo, miembro, presidente, cargo, comercial, promedio, fecha, santiago, carta
Topic #2:	gestión, corporativo, información, valor, proveedor, grupo, sostenibilidad, desempeño, interés, tema, riesgo, cada, responsable, negocio, financiero

Tópico	Keywords
Topic #3:	riesgo, financiero, contar, mercado, ambiental, nuevo, comunidad, inversión, filial, participación, actividad, social, nivel, crédito, relacionado
Topic #4:	desarrollo, trabajo, persona, trabajador, resultado, seguridad, salud, línea, desarrollar, distinto, primero, trabajar, buscar, cultura, mejor
Topic #5:	transformación, total, anual, servicio, digital, millón, acuerdo, control, mujer, hombre, unidad, activo, tiempo, herramienta, región
Topic #6:	centro, residuo, derecho, nacional, dar, canal, diciembre, junto, fin, cuenta, partir, humanos, comercial, san, escuela
Topic #7:	agua, energía, cambio, nuevo, emisión, consumo, generación, crecimiento, día, sostenible, sistema, uso, ambiental, servicio, eléctrico

Realizando ese análisis llegamos a los siguientes nombres en orden de aparición:

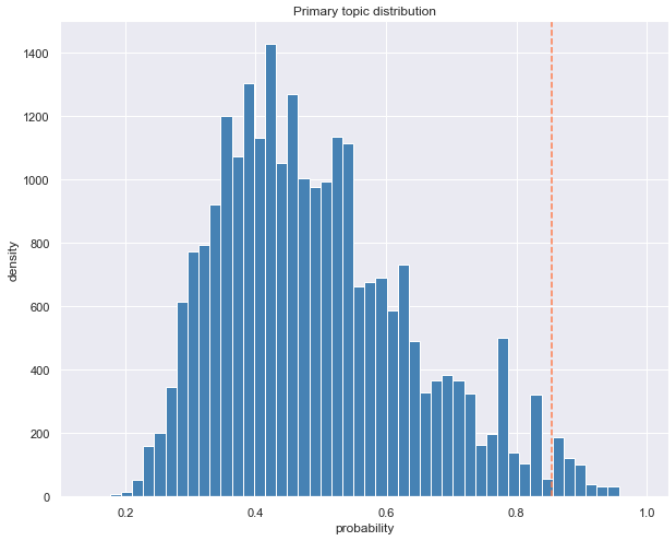
"Mirada sostenible desde el directorio", "Creación de valor sustentable", "Acciones sostenibles que impactan el valor financiero", "Sustentabilidad a partir de colaboradores", "Inclusión e Innovación", "Derechos humanos", "Medio ambiente".

Con la información obtenida por los tópicos realizamos una tabla de frecuencia cruzada. De esta manera podemos ver el solapamiento de tópicos por empresa y ver cuales hablan sobre los mismos tópicos con mayor frecuencia, cuales tópicos tienen mayor cobertura y cuales corresponden a nichos.



Se normaliza la frecuencia de tópicos entre 0 y 1 considerando el porcentaje de menciones para minimizar el problema descrito en un comienzo, la diferencia de cantidad de oraciones por empresa.

Luego calculamos la distribución de probabilidad, es decir, la probabilidad de que una oración pertenece al tópico asignado. Con esto visualizamos que tan seguro está nuestro modelo al clasificar cada oración dentro de un tópico. Así, definimos un threshold que servirá para extraer las principales oraciones que definen el enfoque de cada empresa.



Pues aquellas oraciones que tengan una asignación con probabilidad mayor al umbral elegido se consideran oraciones de interés, que potencialmente definen tanto a los tópicos como los temas de mayor influencia en cada empresa.

Considerando el tópico “Sustentabilidad a partir de colaboradores” se obtienen los siguientes resultados:

company	probability	sentences
aguas andinas	0.952372	Compromiso Promover la diversidad y el bienestar en el trabajo garantizando la seguridad y salud laboral favoreciendo el desarrollo y la promoción del talento e impulsando una cultura colaborativa e innovadora
sonda	0.938759	Enfocamos el desarrollo de talentos para generar equipos de trabajo y profesionales ágiles y preparados para entregar soluciones acordes a las industrias de nuestros clientes
cencosud	0.934051	La Compañía aspira a que sus colaboradores se sientan orgullosos de ser parte de Cencosud así como promover un clima de trabajo donde prime la confianza el respeto mutuo y la inclusión
cencosud	0.934045	Para lograr lo anterior se desarrollan distintas iniciativas que se llevan a cabo con ellos con el objetivo de apoyar su desarrollo y potenciar el trabajo conjunto
itaucorp	0.928563	Además refuerza la diversidad que promueve el banco donde el mejor argumento es el que vale y la única jerarquía que importa es la de la mejor idea
itaucorp	0.928397	Diálogo y transparencia Construir relaciones de confianza y permanentes para mejorar nuestros negocios y generar valor compartido
aguas andinas	0.922002	Cuatro grupos focales con mujeres para diseñar e implementar medidas de equidad de género en la organización
aguas andinas	0.921989	Compromiso Favorecer la mejora en la calidad de vida de los ciudadanos y promover la creación de entornos saludables

De manera similar podemos agrupar por empresa, analizando sus principales oraciones por cada tópico.

Tomando Falabella como ejemplo se obtienen los siguientes resultados:

Tópico	Oración	Lematización
Mirada sostenible desde el directorio	SUPERMERCADOS Tottus opera hipermercados supermercados y formatos de precio conveniente Hiperbodega Precio Uno	supermercado tottus operar hipermercado supermercado formato precio conveniente hiperbodega precio
Acciones sostenibles que impactan el valor financiero	Esta división incluye CMR Falabella tarjeta de crédito Banco Falabella banco Seguros Falabella corredora de seguros y CF Seguros compañía de seguros	división incluir cmr falabello tarjeta crédito banco falabello banco seguro falabella corredora seguro cf seguro compañía seguro
Inclusión e Inovación	puntos de CC son multiformato donde nuestros clientes pueden comprar sus productos en Falabella	punto cc ser multiformato cliente poder comprar producto falabella
Sustentabilidad a partir de colaboradores	Se trabajó conjuntamente con los sindicatos para implementar medidas de apoyo y tranquilidad a todos los colaboradores	trabajar conjuntamente sindicato implementar medida apoyo tranquilidad colaborador
Creación de valor sustentable	A su vez en enero de aprobamos políticas corporativas de prevención de delitos y antisoborno y un texto actualizado del Modelo de Prevención de Delitos	vez enero aprobar política corporativo prevención delito antisoborno texto actualizado modelo prevención delito
Medio ambiente	Con ello la proporción de energías renovables autogeneradas se elevará hasta un del total de electricidad consumido por la organización	proporción energía renovable autogenerado elevar total electricidad consumido organización
Derechos humanos	Esta auditoría fue desarrollada de acuerdo con el International Standard on Assurance Engagement ISAE Assurance Engagements other than Audits or Reviews of Historical Financial Information	auditoría ser desarrollado acuerdo international standard on assurance engagement isae assurance engagement other than audits or review of historical financial information

Mostrando las principales oraciones emitidas por el reporte de Falabella, para cada tópico encontrado en nuestro modelo.

Si bien algunas puede que no calcen como se esperaba puede que valga la pena evaluar un top dentro de las primeras oraciones con mayor relevancia por tópico, para eliminar clasificaciones erradas por el modelo, que no es perfecto, y tener una idea más general sobre las principales acciones y posturas en cada tema.