Predicción de la Respuesta en un Sistema de Búsqueda de Respuesta Semántico.

Matías Oyarzun and Sandra Roger

Grupo de Investigación en Lenguajes e Inteligencia Artificial (GILIA),
Facultad de Informática. Universidad Nacional del Comahue
Buenos Aires 1400, (8300) Neuquén
matias.oyarzun@est.fi.uncoma.edu.ar
roger@fi.uncoma.edu.ar

Resumen En este artículo se describe un primer prototipo que se ha desarrollado para la tarea de Predicción de la Categoría planteada en el desafío SMART[1]. Este problema se puede plantear como una tarea de clasificación multiclase, pues toma preguntas en lenguaje natural y devuelve la categoría (resource, literal, boolean) a la que pertenecen. Para el entrenamiento, se utilizaron los datasets de DBPedia y Wikidata de los SMART 2020 y 2021. En este prototipo, se entrenaron 4 modelos de aprendizaje automático con distintas combinaciones de los datasets para hallar el más preciso. El mejor modelo, obtuvo una precisión del 97, 2 % y 96,8 % para los datasets de DBpedia y Wikidata, respectivamente. En ambos casos, se utilizó el clasificador Support-Vector Machines (SVM). Posteriormente, se busca también la construcción de un modelo para la tarea de Predicción del Tipo de Respuesta. Ésto permitirá, finalmente, la implementación de un Sistema de Búsqueda de Respuestas eficiente.

Keywords: Clasificación de Preguntas, Aprendizaje Automático, SMART, Question Answering.

Contexto Este trabajo está parcialmente financiado por la UNCo, en el marco del nuevo proyecto de investigación *Tecnologías Semánticas para el desarrollo de Agentes Inteligentes*. Como así también, lo financia parcialmente el Consejo Interuniversitario Nacional (CIN) con una Beca de Estímulo a las Vocaciones Científicas 2021. Este trabajo formará parte de la propuesta de tesis final de carrera.

1. Introducción

El proceso de QA (*Question Answering* - QA) consta de una etapa de análisis de la pregunta, recuperación de los datos relevante de fuentes de conocimiento y la extracción de la información concreta y correcta como respuesta.

El análisis de la pregunta es fundamental. En este sentido, continuando con [4] nos concentramos, en una primera etapa, en la predicción de la respuesta esperada a partir de la pregunta de entrada. En este sentido, se ha realizado un análisis

[1] https://smart-task.github.io/

de las diferentes metodologías y estudio de herramientas disponibles para realizar una implementación eficiente.

Dentro de la tarea de QA, focalizada en la predicción del tipo de respuesta, existen distintas competencias, una de ellas es el desafío denominado SMART *SeMantic Answer Type and Relation Prediction Task*, de la cual se han realizado hasta el momento dos instancias de tales competencias en los años 2020 [2] y 2021, y una última que se encuentra en proceso de este año.

Para el desafío, es posible realizar una clasificación del tipo de respuesta granular con ontologías de Web Semántica populares como DBpedia (∼760 clases) y Wikidata (∼50K clases). En esta competencia se cuenta con dos tareas principales e independientes: 1) predicción del tipo de respuesta y 2) predicción de un conjunto de relaciones usadas para la identificación de la respuesta correcta.

La primer tarea, predicción del tipo de respuesta, consiste en la predicción de la "categoría" (*resource, literal* y *boolean*) y la predicción del "tipo de la respuesta". En el caso de que la respuesta sea *resource*, el tipo de la respuesta son clases de ontologías. Si es literal, entonces el tipo de la respuesta puede ser un número, una fecha o una cadena. Finalmente, si es *boolean*, el tipo de la respuesta es siempre *boolean*.

La tarea de predicción de relaciones para la pregunta es una tarea difícil: algunas relaciones están alejadas semánticamente, a veces los tokens que deciden las relaciones están distribuidas a lo largo de la pregunta, algunas relaciones están implícitas en el texto, entre otras.

Se implementó un módulo para la clasificación del tipo de respuesta utilizando aprendizaje automático. Para lograr ésto, la competencia dispone de varios corpus que se emplearon para el entrenamiento y testeo de la clasificación de la categoría y el tipo de respuesta para cada una de las diferentes ontologías que proponen. Se llevó a cabo el entrenamiento y testeo de diversos modelos de aprendizaje automático, entre ellos se encuentran *Support-Vector Machine, Logistic Regression, Naive Bayes y Decision Tree*. A partir de la precisión de cada uno de ellos, se efectuó una comparación para hallar el mejor modelo.

Asimismo, se pretende diseñar y desarrollar un módulo para la segunda tarea de predicción de relaciones usando tanto la ontología de DBpedia como la de Wikidata. Al igual que en la primera tarea, se provee de corpus para trabajar.

2. Nuestra Propuesta

2.1. Preprocesamiento de los Datos

Antes de la construcción de los modelos de aprendizaje automático, se llevó a cabo un preprocesamiento de los datos, donde se realizó una limpieza y transformación de los mismos para un formato deseado.

Tanto *datasets* de entrenamiento como los de testeo son provistos por la competencia SMART, ambos en formato JSON. Ésto permitió cargar los mismos utilizando una representación tabular, lo que nos proporciona una fácil manipulación de los datos.

Para lograr una construcción robusta de los modelos de aprendizaje, se eliminaron aquellas filas que no presentaban valores o que contenían valores inválidos. A su vez, se eliminaron de las preguntas aquellas palabras que no eran alfanuméricas. Posteriormente, se realizó una tokenización de las preguntas, donde se dividió cada pregunta en partes más pequeñas llamadas tokens (cada palabra de la misma). Estos tokens se utilizaron para realizar una normalización del texto, en la cual se efectuó el *stemming* y lematización. El *stemming* es el proceso de reducir la inflexión de las palabras a sus formas raíz, incluso si la propia raíz no es una palabra válida en la lengua. Mientras que la lematización, reduce las palabras inflexionadas asegurándose de que la palabra raíz pertenece a la lengua, esta raíz se llama lema y es la forma canónica de un conjunto de palabras.

Inicialmente, en este estudio se buscaba darle un trato especial tanto a las partículas interrogativas de las preguntas como a las stopwords, por lo que se decidió adaptar los *datasets* para lograr ello. Sin embargo, estos nuevos *datasets* funcionaban con menos eficacia. Por lo tanto, se optó por dejar las partículas interrogativas y las *stopwords* tal cual provienen del *dataset* original.

2.2. Selección de Características

Efectuar una buena selección de características para los modelos de aprendizaje, aporta ciertos beneficios al proceso de aprendizaje, pues reduce la dimensionalidad, eliminación de ruido, entre otros. Todo esto permite mejorar la velocidad de cómputo a la hora de entrenar y evaluar el clasificador, y hasta evitar el *overfitting.*

Para lograr independizar el modelo a utilizar con respecto a los recursos lingüísticos, se analizó la frecuencia de términos (TF) y la frecuencia de términos inversa (TF-IDF) en la extracción de características utilizando CountVectorizer y TFIDFVectorizer. La evidencia empírica concluye que aplicar TFIDFVectorizer junto con unigramas y bigramas es la opción más adecuada para esta tarea.

El *target* de esta tarea, la categoría, se encuentra en forma de cadena en los *datasets* originales. Sin embargo, se determinó transformar en una etiqueta numérica de tal manera que permita a los modelos trabajar correctamente.

2.3. Diseño de los Modelos de Aprendizaje

Para comenzar con los experimentos iniciales, se optó por seguir un enfoque en dos fases. En la primer fase, se realizó la clasificación de las categorías de las preguntas. Posteriormente, para la segunda fase, se pretende diseñar un modulo para la predicción de tipos de las preguntas para aquellas para las que se predijo que la categoría era recurso o literal.

Para la clasificación de categorías se realizaron pruebas sobre 4 tipos de clasificadores: *Naive Bayes* (NB), *Support-Vector Machine* (SVM), *Decision Tree* (DT) y *Logistic Regression* (LR), pues eran los más utilizados por los participantes de la competencia [1] [5] [3]. Para los modelos se utilizó el *dataset* provisto por SMART tanto para el año 2020 como el 2021, efectuándose pruebas con los mismos por separado y combinándolos. En cada prueba se hacían cambios en la

Tabla 1: Dataset provisto por SMART para el año 2020

Dataset	Preguntas		Categorías		
	Train	Test	Boolean	Literal	Resource
DBpedia	17,571	6,883	2,799	5,188	9,584
Wikidata	18,251	4,571	2,139	4,429	11,683

Tabla 2: Dataset provisto por SMART para el año 2021

Dataset	Preguntas		Categorías		
	Train	Test	Boolean	Literal	Resource
DBpedia	29,336	7,334	1,794	3,363	24,179
Wikidata	34,843	8,711	1,693	3,663	29,487

Tabla 3: Dataset combinado 2020-2021

Dataset	Preguntas		Categorías		
	Train	Test	Boolean	Literal	Resource
DBpedia	39,512	4,381	2,525	5,147	31,840
Wikidata	44,688	4,571	2,556	5,303	38,855

parametrización de cada uno de los modelos, tanto de las características utilizadas para el entrenamiento como de los parámetros recibidos. A través de estas pruebas, se logró obtener un mejor modelo, en el cual se utilizaron unigramas y bigramas de palabras ponderados por TFIDF como características para entrenar un clasificador SVM con un kernel lineal y sus parámetros por defecto.

3. Resultados Experimentales

3.1. *Datasets*

La Tabla 1 y la Tabla 2 presentan las estadísticas descriptivas para los dos conjuntos de datos, DBpedia y Wikidata, provistos por la competencia SMART para los años 2020 y 2021 respectivamente. Wikidata tiene un poco más de recursos que tipos de respuestas literales, en comparación con DBpedia. A su vez, la Tabla 3 nos presenta el resultado de combinar los *datasets* de DBpedia y Wikidata para los años 2020 y 2021.

A continuación se expondrán los resultados obtenidos en distintas pruebas realizadas. Particularmente, la tarea de clasificación de la categoría fue evaluada en términos de la precisión de la clasificación.

La Tabla 4 presenta los resultados obtenidos a partir de la experimentación realizada al *dataset* por separado correspondiente a los años 2020 y 2021 respectivamente. Como se puede apreciar la mejor precisión es obtenida para ambos años es con el método de SVM tanto para del *dataset* de DBPedia como para Wikidata. Si analizamos los resultados de la del *dataset* combinado de ambos años la misma tendencia se mantiene para el caso de DBPedia, pero no ocurre lo mismo para el caso de Wikidata donde la mejor precisión se obtuvo para el método DT.

Tabla 4: Resultados *dataset* 2020, 2021 y combinando ambos años

2020			**2021**			**2020-2021**		
Dataset	**Método**	**Train**	**Dataset**	**Método**	**Train**	**Dataset**	**Método**	**Train**
DBpedia	LR	0.931	DBpedia	LR	0.958	DBpedia	LR	0.906
	SVM	**0.945**		**SVM**	**0.972**		**SVM**	**0.939**
	NB	0.889		NB	0.938		NB	0.850
	DT	0.913		DT	0.961		DT	0.911
Wikidata	LR	0.922	Wikidata	LR	0.956	Wikidata	LR	0.925
	SVM	**0.936**		**SVM**	**0.968**		SVM	0.963
	NB	0.885		NB	0.943		NB	0.902
	DT	0.914		DT	0.960		**DT**	**0.972**

4. Conclusiones

A partir de los datos de DBpedia y Wikidata provistos por SMART, se llevaron a cabo varias pruebas sobre la normalización del texto, desde darle un trato especial a las partículas interrogativas hasta remover las stopwords de las preguntas. De la misma manera, se realizaron pruebas sobre distintos modelos para encontrar el más adecuado para el problema de clasificación.

En otros estudios [5] [3] realizados por participantes del desafío SMART, se utilizaron los modelos BERT ajustados. Sin embargo, debido a las limitaciones en los recursos informáticos, se utilizaron los modelos mencionados para tener un coste computacional barato durante el desarrollo. Esto abre la puerta a futuras mejoras, en la cual se podría utilizar redes neuronales o enfoques híbridos para lograr una mayor eficacia, así como intentar aumentar los *datasets* de entrenamiento para lograr un balanceo en la cantidad de categorías existentes de cada tipo.

Asimismo, se pretende diseñar y desarrollar un prototipo para la tarea de predicción de relaciones usando ambas ontologías, y con ello finalizar las principales tareas del desafío.

Referencias

1. C. Kim and E. Jimenez-Ruiz. CitySAT: a System for the Semantic Answer Type Prediction Task. In *CEUR Workshop Proceedings*, volume 3119, pages 77–88. CEUR, 2022.
2. N. Mihindukulasooriya, M. Dubey, A. Gliozzo, J. Lehmann, A.-C. N. Ngomo, and R. Usbeck. SeMantic AnsweR Type prediction task (SMART) at ISWC 2020 Semantic Web Challenge. *CoRR/arXiv*, abs/2012.00555, 2020.
3. C. Nikas, P. Fafalios, and Y. Tzitzikas. Two-stage Semantic Answer Type Prediction for Question Answering using BERT and Class-Specificity Rewarding. In *SMART@ ISWC*, pages 19–28, 2020.
4. M. Oyarzun and S. Roger. Tecnologías de sistemas de QA aplicadas a la Web Semántica. In *XXIII Workshop de Investigadores en Ciencias de la Computación (WICC 2021, Chilecito, La Rioja)*, 2021.
5. V. Setty and K. Balog. Semantic Answer Type Prediction using BERT: IAI at the ISWC SMART Task 2020. *arXiv preprint arXiv:2109.06714*, 2021.