Segundo Mini Projeto de Língua Natural

Grupo 21

Duarte Teles 83450 | Ricardo Brancas 83557 | Daniel Oliveira 87848

26 de Maio de 2019

1 Introdução

O objetivo deste projeto é desenvolver uma métrica de similaridade que permita classificar, em relação ao tipo, questões dadas sobre cinema. Por exemplo: a classificação da pergunta *"What are the most relevant actors in Bad Boys?"* é `actor_name`.

Para o desenvolvimento da nossa solução, utilizaremos um *corpus* pré-etiquetado (`QuestoesConhecidas.txt`) para treinar o nosso classificador; um conjunto de *embeddings* pré-treinados (ver secção 2.2) e ainda um *corpus* de desenvolvimento (`NovasQuestoes.txt`).

2 Arquitetura do Modelo

2.1 Pré-Processamento

O nosso corpus de treino consiste num conjunto de pares $\langle \text{classificação}, \text{pergunta} \rangle$. De modo a minimizar o ruído provocado por partes pouco relevantes da frase, realizamos vários passos de pré-processamento, tanto para o *corpus* de treino, como para o novo conjunto de questões a classificar:

Por cada um dos ficheiros da pasta `recursos` (com a exceção do ficheiro `list_keywords.txt`), substituimos qualquer ocorrência das palavras contidas no mesmo por uma única palavra genérica. Por exemplo, todos os títulos de filmes conhecidos são substituídos pela palavra `movie`. Isto impede o classificador de fazer *overfitting* aos nomes próprios contidos no *corpus* de treino. Excluímos o ficheiro `list_keywords.txt` porque continha várias palavras que são importantes para a nossa tarefa, como por exemplo a palavra `actors`.

De seguida segmentamos a frase obtida, removendo toda a pontuação e números existentes, obtendo uma lista de *tokens* que é depois filtrada com recurso às *stop words* do `NLTK`. Estas *stop words* consistem num conjunto de palavras comuns que para a nossa tarefa não possuem significado relevante. Alguns exemplos: `me`, `you`, `that'll`, `now`.

Por fim eliminamos todas as palavras para as quais não possuímos um *embedding*, visto que não seria possível relacioná-las de qualquer modo.

2.2 Obtenção dos *Embeddings*

As palavras do vocabulário de um sistema de Língua Natural podem ser representadas de várias formas e, dependendo dessa representação, pode ou não ser possível estabelecer-se uma relação semântica entre diferentes palavras.

Uma possível representação para os nossos símbolos (palavras) que permite estabelecer relações de semântica é a utilização de *Vector Space Models* [3]. A representação de palavras num espaço vetorial contínuo permite mapear palavras com significado semântico semelhante em zonas próximas do espaço. Esta assunção é baseada na *Distributional Hypothesis* [2] que afirma que palavras que ocorram em contextos semelhantes têm semântica semelhante.

As abordagens que utilizam este principio são *count based methods* e *predictive methods*. O primeiro calcula a co-ocorrência de palavras com os seus vizinhos e mapeia estas contagens num vetor denso para cada palavra. Os modelos preditivos tentam prever diretamente a ocorrência de uma palavra dado os vizinhos de acordo com os vários contextos em que cada palavra ocorre.

Assim, de modo a extrair semântica de cada frase e relacionar essa semântica com os tópicos correspondentes, utilizamos um modelo de *embeddings* [4] obtidos através de `WordToVec` [5].

Este modelo baseia-se no método preditivo, em particular na arquitetura de modelo *Continuous Bag of Words* [5], no qual a ordem das palavras não importa e onde existe uma janela sobre a frase através da qual as palavras vizinhas são utilizadas para tentar estimar a melhor codificação de forma a prever cada palavra no corpus

de treino, utilizando os múltiplos contextos em que a mesma ocorre.

2.3 Semântica da Frase

No modelo de *embeddings* que estamos a utilizar, o conteúdo semântico está ao nível da palavra. Para extrair semântica da frase, com o modelo aqui apresentado, é necessário combinar os *embeddings* das palavras da frase de algum modo. Uma das formas mais simples de o fazer consiste apenas em somar os *embeddings* das diferentes palavras e depois normalizar, o que dará origem a um único vetor para cada frase.

Esta abordagem tem a vantagem de ser mais simples que outras apresentadas no trabalho futuro ao nível da complexidade computacional, mas traz também alguns problemas. Em particular, qualquer palavra ou carácter na frase, desde que tenha um *embedding* pré treinado, irá ter um peso equivalente a qualquer outra palavra. Isto é mitigado através do pré-processamento realizado, mas seria útil ter a possibilidade de pesar as palavras de uma forma mais generalizada.

2.4 Medida de Similaridade

Como medida de similaridade entre frases (representadas como vetores) foi utilizado o cosseno. Este funciona como uma distância euclidiana normalizada, a qual pode ser utilizada no nosso problema pois os valores dos vetores estão normalizados entre -1 e 1.

3 Análise de Resultados

3.1 Ferramentas

Para obter os resultados apresentados de seguida utilizámos as seguintes ferramentas:

- O pacote `NLTK` e, em particular, o módulo `stopwords` cuja utilização foi descrita na secção 2.1.
- O pacote `gensim`, utilizado para carregar o modelo e obter o *embedding* das diferentes palavras.
- O pacote `scikit-learn`, utilizado para facilitar alguns cálculos matemáticos.
- O modelo de *embeddings* treinado sobre o *corpus* "Google News" usando *Continuos Bag of Words* com vetores de dimensão 300, já referido na secção 2.2.

3.2 Resultados

Na figura 1 apresentamos os resultados obtidos da classificação do *corpus* de teste fornecido; como é possível observar, o nosso classificador atribui categorias corretas para todas as questões, obtendo uma acurácia de 100%.

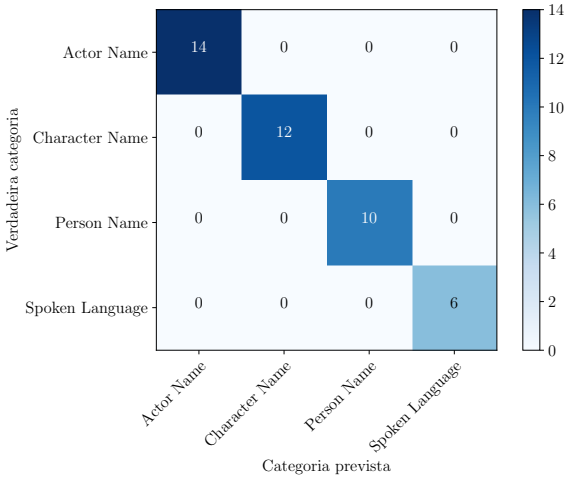

Figura 1: Matriz de confusão para o *corpus* de teste fornecido.

Já na figura 2 apresentamos os resultados que obtivemos utilizando um *corpus* de teste estendido por nós. Considerando este conjunto de perguntas mais amplas, incluindo algumas paras as quais temos muito poucos exemplos, a acurácia obtida é 98%.

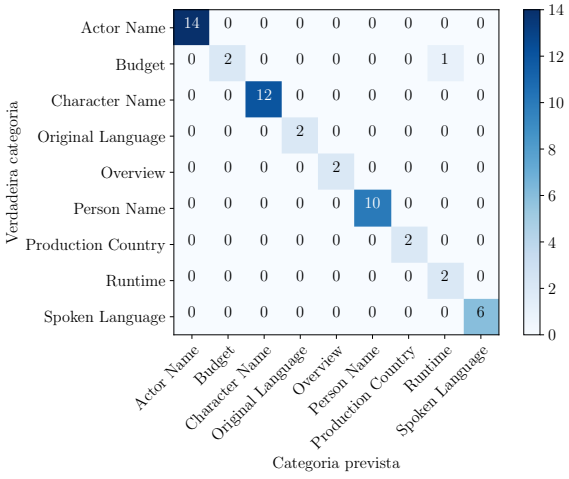

Figura 2: Matriz de confusão para um *corpus* de teste estendido.

Experimentámos ainda usar apenas o *subset* do modelo de *embeddings* incluído com o `NLTK`, obtendo uma acurácia de 97% no *corpus* de desenvolvimento original e cerca de 94% no estendido.

4 Trabalho Futuro

De forma a melhorar a prestação do nosso modelo e a resolver o problema que a abordagem *Continuous Bag of Words* nos traz, podemos recorrer a modelos que pesam de forma independente a presença de cada tópico na frase. Um modelo a considerar é o *Latent Dirichlet Allocation* [1], que descreve a distribuição de tópicos por documento. Poderíamos ainda utilizar outras abordagens para resolver o problema tal como *Text Frequency-Inverse Document Frequency*.

Referências

[1] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. In *Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada]*, pages 601–608, 2001.

[2] Melody Dye, Michael N. Jones, Daniel Yarlett, and Michael Ramscar. Refining the distributional hypothesis: A role for time and context in semantic representation. In *Proceedings of the 39th Annual Meeting of the Cognitive Science Society, CogSci 2017, London, UK, 16-29 July 2017*, 2017.

[3] Katrin Erk and Sebastian Padó. A structured vector space model for word meaning in context. In *2008 Conference on Empirical Methods in Natural Language Processing, EMNLP 2008, Proceedings of the Conference, 25-27 October 2008, Honolulu, Hawaii, USA, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 897–906, 2008.

[4] Pre-trained word vectors on google news dataset. `https://code.google.com/archive/p/word2vec/`.

[5] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.