Nome: Rafael Claro Ito (R.A.: 118430)
Resumo do artigo: A Unified Architecture for Natural Language Processing: Deep Neural Networks with Multitask Learning

Neste artigo, os autores descrevem uma arquitetura de rede neural convolucional que resolve diversos problemas de forma conjunta, tratados até a publicação do artigo comumente como separados. Esse processo, chamado de *multitask learning (MTL)*, aprende diversas tarefas ao mesmo tempo com o objetivo de benefícios mútuos. A rede recebe como entrada uma determinada frase e gera saídas para as seguintes tarefas:

- ***part-of-speech tagging (POS)***: rotula cada palavra de acordo com sua função sintática;
- ***chunking***: rotula segmentos de uma frase com constituintes sintáticas;
- ***named entity tags (NER)***: rótulos para classificar objetos do mundo real, como pessoas, organizações, lugares, etc;
- ***semantic roles labelling (SRL)***: atribui um rótulo aos constituintes sintáticos da frase de acordo com a função semântica;
- ***language model***: tradicionalmente, estima a probabilidade da próxima palavra, dada uma sequência de palavras. Entretanto, aqui o modelo de linguagem será empregado para classificar se um texto é real ou gerado artificialmente.
- ***semantically similar words***: tarefa que prediz a relação semântica de duas palavras (ex: sinônimas);

Para isso, a rede toda é treinada nas diversas tarefas de forma conjunta. A tarefa do modelo de linguagem é treinada de forma não-supervisionada usando o site inteiro da *Wikipedia* como dataset, enquanto que todas outras tarefas são treinadas de forma supervisionada, considerando o critério entropia cruzada. Tal abordagem é tratada como aprendizado semi-supervisionado.

A tarefa principal do problema (e também a mais desafiadora) é a *SRL*. Utilizou-se todas as outras tarefas para mostrar a generalidade da arquitetura proposta e também para melhorar a resolução do problema *SRL* através do *multitask learning*.

O modelo proposto nesse artigo emprega o uso de uma rede neural profunda para extrair as *features* relevantes às tarefas. A primeira camada extrai *features* das palavras, mapeando-as em vetores reais (*lookup-table*). A segunda camada extrai *features* das frases tratando-a como uma sequência com estruturas local e global (ao contrário da abordagem *bag-of-words*). As camadas seguintes são camadas comuns de uma rede neural.

Em uma rede neural tradicional o número de entradas é fixo. No entanto, como o número de palavras em frases é variável, a solução mais simples é trabalhar com janelas de tamanhos fixos que “varrem” a frase. Embora a abordagem de janelas possa ser útil para tarefas como *POS*, para tarefas mais complexas como *SRL* ela se torna inviável. Isso ocorre pois ao se analisar funções semânticas, palavras fora da janela podem ser importantes para o contexto. Assim, para casos onde as dependências de longa distâncias são importantes, opta-se pelas redes *TDNNs* (*Time-Delay Neural Networks*), pois elas consideram todas as janelas na frase. Na saída da *TDNN* usa-se uma camada *Max Over Time*, responsável por capturar as *features* mais relevantes da frase. As próximas camadas são de uma rede neural usual. Elas são importantes para adicionar não-linearidades ao modelo, essencial para a tarefa *SRL* (a *TDNN* realiza operações lineares). Por fim, o número de neurônios na penúltima camada é igual ao número de classes para a tarefa em questão, sendo a última camada a função *softmax*.

Nas seções 4 e 5, várias comparações são feitas a respeito de trabalhos anteriores nas áreas de *multitask learning* para *NLP*, modelos de linguagem e aprendizado semi-supervisionado.

A última seção antes da conclusão compara os resultados obtidos dependendo das tarefas treinadas. O modelo de linguagem foi inicialmente treinado sozinho, sendo que a *lookup-table* obtida foi usada para inicializar a *lookup-table* no *MTL*. Para a tarefa *SRL*, obteve-se uma taxa de erro por palavra de 14.3%, caracterizando o modelo como *state-of-the-art*. Todos os experimentos de *MTL* tiveram performance maior do que o modelo treinado apenas para *SRL*, mostrado de forma bem evidente na figura 3. Para *POS* e *chunking* os resultados em *MTL* apresentam melhoras, mas não tão significativas. Tanto para estas tarefas quanto para *SRL* o tempo de teste por frase é de alguns milésimos de segundo.

Por fim, o artigo termina exaltando a rapidez com que a arquitetura processa frases, afirmando que aprender tarefas simultaneamente pode melhorar a performance de generalização. Também é constatado que para a tarefa de *SRL* o modelo atingiu performance *state-of-the-art*. O mais impressionante é que tudo isso foi obtido sem passar ao modelo qualquer *feature* com informação de sintaxe, o que se considerava obrigatório.