Proposta recebida em Setembro 2016 e aceite para publicação em Novembro 2016.

FlexSTS: Um Framework para Similaridade Semântica Textual

FlexSTS: A Framework for Semantic Textual Similarity

Jânio Freire
Universidade de Fortaleza
janio.freire@gmail.com

Vládia Pinheiro
Universidade de Fortaleza
vladiacelia@unifor.br

David Feitosa
Universidade de Fortaleza
davidfeitosa@gmail.com

Resumo

Desde 2012, os eventos de *Semantic Evaluation* (SemEval) propõem a tarefa de Similaridade Semântica Textual (STS) como um tema de competição, demonstrando sua relevância. Em 2016, a tarefa foi, pela primeira vez, proposta para língua portuguesa, no Workshop de Avaliação de Similaridade Semântica e Inferência Textual (ASSIN), realizado durante a conferência PROPOR 2016. Neste trabalho, apresentamos o FlexSTS — um *framework* flexível para STS que combina diversos componentes como parsers morfológicos e sintáticos, bases de conhecimento e lexicais, algoritmos de aprendizagem automática, e algoritmos de alinhamento e cálculo da similaridade. Para a ASSIN, FlexSTS foi instanciado em três sistemas de STS para língua portuguesa. Os resultados obtidos foram comparados com uma abordagem *baseline* que utiliza o coeficiente DICE.

Palavras chave

Similaridade Textual, Similaridade Semântica, Avaliação Semântica

Abstract

Since 2012, Semantic Evaluation series (SemEval) propose the task of Semantic Textual Similarity (STS) as a evaluation theme, demonstrating the relevance of this research topic. In 2016, the task was first proposed to the Portuguese language, in the Workshop of Semantic Textual Similarity and Inference Evaluation (ASSIN), held during the conference PROPOR 2016. In this paper, we present the FlexSTS — a flexible framework for STS combining several components as morphological and syntactic parsers, knowledge and lexical databases, machine learning algorithms, and algorithms for alignment and similarity. For ASSIN, FlexSTS was instantiated into three STS systems for Portuguese. The results were compared with a baseline approach that uses DICE coefficient.

Keywords

Textual Similarity, Semantic Similarity, Semantic Evaluation.

1 Introdução

A tarefa de Similaridade Semântica Textual (STS) (Agirre et al., 2013) visa medir o grau de equivalência semântica entre dois textos, capturando a noção de que alguns textos são mais similares que outros. Por exemplo, o par de sentenças "*A organização criminosa é formada por diversos empresários e por um deputado estadual*" e "*Segundo a investigação, diversos empresários e um deputado estadual integram o grupo.*" devem receber um valor de similaridade mais alto que o par de sentenças "*Mas esta é a primeira vez que um chefe da Igreja Católica usa a palavra em público.*" e "*A Alemanha reconheceu ontem pela primeira vez o genocídio armênio*". STS difere das tarefas de Inferência textual (RTE) e Detecção de Paráfrase, principalmente por assumir uma equivalência bidirecional.

Computar a similaridade textual é útil para um número crescente de tarefas de Processamento de Linguagem Natural (PLN) e Inteligência Artificial (IA), tais como a sumarização (Lin & Hovy, 2003) ou o reuso de experiência (Albuquerque et al., 2012).

Desde 2012, os eventos de *Semantic Evaluation* (SemEval)[1] propõem esta tarefa como um tema de competição, demonstrando a relevância da mesma e um tema de pesquisa ainda em aberto. Em 2016, a tarefa foi novamente proposta para língua inglesa na edição do SemEval 2016[2] e, de forma inédita para língua portuguesa, no Workshop de Avaliação de Similaridade Semântica e Inferência Textual (ASSIN), realizado durante a conferência PROPOR 2016[3].

Tradicionalmente, a tarefa consiste em computar o grau de similaridade semântica entre duas sentenças, usando a seguinte escala:

1. Sentenças completamente diferentes, em assuntos diferentes;

[1] https://en.wikipedia.org/wiki/SemEval
[2] http://alt.qcri.org/semeval2016/task1/
[3] http://propor2016.di.fc.ul.pt

2. Sentenças não relacionadas, mas que compactuam do mesmo assunto;
3. Sentenças de certa forma relacionadas, que podem descrever fatos diferentes mas compartilham alguns detalhes;
4. Sentenças fortemente relacionadas, que divergem apenas em alguns detalhes;
5. Sentenças significam exatamente a mesma coisa.

Neste trabalho, apresentamos o FlexSTS — um *framework* genérico que facilita e flexibiliza o desenvolvimento de sistemas de STS, pois combina diversos componentes como *parsers* morfológicos e sintáticos (NLP toolkits), bases de conhecimento e lexicais, algoritmos de aprendizagem automática, e algoritmos de alinhamento e cálculo da similaridade. Especificamente para avaliação no Workshop ASSIN, FlexSTS foi instanciado para língua portuguesa em três configurações (sistemas) usando o parser *Freeling* (Padró & Stanilovsky, 2012), o modelo de similaridade entre palavras HAL (Hyperspace Analog to Language) (Burgess et al., 1998), a base de conhecimento Wordnet (Miller, 1995), o algoritmo de aprendizagem automática proposto por Pedregosa et al. (2011), e o modelo de alinhamento entre termos proposto por Han et al. (2013). Foram enviadas as execuções dos três sistemas de STS e os resultados obtidos foram comparados com uma abordagem *baseline* que utiliza o coeficiente DICE (Rohlf, 1992) de similaridade sintática entre textos. A análise de casos em que nosso melhor sistema não obteve nível de acerto desejado indiciam melhorias para trabalhos futuros.

2 Trabalhos Relacionados

Destacam-se, como estado da arte, os sistemas campeões da tarefa de STS das edições do SemEval 2013, 2014, 2015.

No SemEval 2013, o sistema campeão foi o submetido pela equipe denominada UMBC (Han et al., 2013). Esse sistema consiste de uma abordagem que agrega conhecimento semântico de uma matriz LSA e da WordNet, além de aplicar uma estratégia de alinhamento e penalização, que determina um conjunto de critérios para um mal alinhamento, e valores e a serem descontados para cada tipo de mal alinhamento. O resultado médio da correlação de Pearson foi 0.6181, para língua inglesa.

Em 2014, a equipe vencedora foi a ECNU (Zhao et al., 2014) que utilizou uma abordagem de aprendizagem de máquina com vários algoritmos e 72 *features*. O algoritmo que obteve melhor resultado foi o *Gradient Boosting*. O resultado médio da correlação de Pearson foi 0,8414, também para língua inglesa.

O sistema campeão da edição de 2015 foi apresentado por Sultan et al. (2015) que propôs uma abordagem de aprendizagem de máquina utilizando o algoritmo *Ridge Requession Model*. As características (*features*) definidas para representar o problema baseiam-se na similaridade entre as sentenças, calculada por uma função que usa uma representação vetorial, criada a partir da matriz LSA, de uma base de paráfrase (Ganitkevitch et al., 2013) e da árvore de dependência sintática. Este sistema obteve resultado de 0,8015 (correlação de Pearson).

3 FlexSTS: *Framework* para Similaridade Semântica Textual

Nesta seção apresentamos a proposta do *framework* FlexSTS, o qual define diversos componentes a serem conectados e usados no desenvolvimento de sistemas de STS, agregando modelos e medidas de similaridade, *toolkits* e algoritmos do estado da arte, em cada etapa do processo de STS. A Figura 1 apresenta o fluxo geral do processo de STS e os diversos componentes ou *plugins* necessários.

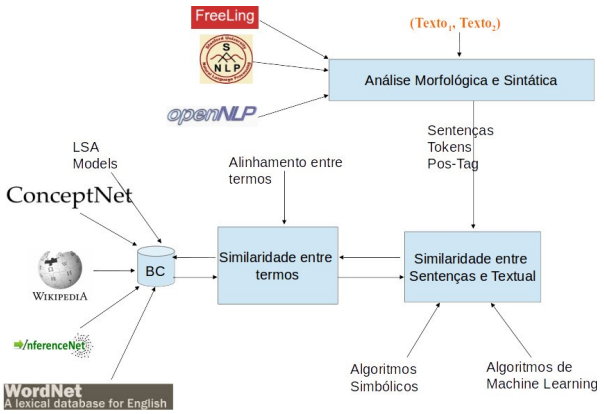

Figura 1: Fluxo do *framework*.

3.1 Análise Morfológica e Sintática

Nesta etapa, dados dois textos de entrada t_1 e t_2, é realizada a detecção das sentenças, a análise morfológica (*tokenização*, lematização, *POS Tagger*) e a análise sintática (*dependency parsing*) de ambos os textos. Inúmeros toolkits disponíveis podem realizar esta tarefa para diversas línguas. Em destaque, tem-se o Stanford NLP Toolkit

(Toutanova et al., 2003), Open NLP (Baldridge, 2005), Freeling (Padró & Stanilovsky, 2012).

O objetivo desta etapa é gerar, para cada texto de entrada, o conjunto de tokens relevantes T_{ij} de cada sentença s_{ij}. O algorıtmo para a construção do conjunto T_{ij}, segue os passos listados abaixo:

1. Análise morfológica e sintática do texto;
2. Reconhecimento de palavras compostas, nomes próprios, valores numéricos, datas e expressões de tempo;
3. Aplicação de heurísticas, seguindo o trabalho de Han et al. (2013):
   (a) Remoção de pontuação;
   (b) Expressões numéricas escritas por extenso são convertidas para números;
   (c) Remoção de *stop words*.
   (d) Referências para tempo são convertidas para o formato militar;
4. Cada *token* das classes abertas de palavras (substantivo, verbo, adverbio e adjetivo), incluindo nomes de entidades reconhecidas, como nomes próprios e abreviações, passam por um processo de desambiguação conforme definido por Pinheiro et al. (2012). Nesse passo, cada termo é associado a um conceito de uma base de conhecimento.
5. Finalmente, o conjunto T_{ij} é formado pelos *tokens* e seus atributos morfológicos, lexicais, sintáticos e semânticos.

3.2 Similaridade Semântica entre Termos

A segunda etapa do processo prevê a aplicação de modelos e medidas para cálculo da similaridade entre palavras $\theta(c, c')$ e de um algoritmo para alinhamento dos termos c e c' de cada sentença s_{1i} e s_{2j} dos textos t_1 e t_2 (textos de entrada).

3.2.1 Modelos de Similaridade Semântica entre Palavras (Word Similarity Models)

O framework define a função $\theta(c, c')$ como uma função parametrizável para vários modelos e medidas de similaridade entre palavras, possibilitando agregar conhecimento adicional expresso em uma ou mais bases de conhecimento e dicionários externos, tais como WikiPedia (Milne & Witten, 2008), WordNet (Miller, 1995), ConceptNet (Liu & Singh, 2004), InferenceNet (Pinheiro et al., 2010a), dentre outras.

Dentre os modelos do estado da arte, tem-se a LSA (*Latent Semantic Analysis*) que segue a hipótese da semântica distribucional, segundo a qual "palavras que ocorrem em contextos similares tendem a ter significados similares" (Harris, 1968). Diversas técnicas de LSA podem ser aplicadas. HAL (*Hyperspace Analog to Language*) (Burgess et al., 1998) é uma técnica de LSA que pode ser aplicada em matriz de co-ocorrência termo-termo. *Singular Value Decomposition* (SVD) (Landauer & Dumais, 1997) tem sido efetiva para melhorar medidas de similaridade entre palavras, visto que podemos selecionar os $k-$maiores valores singulares e reduzir para tamanho k o vetor que representa uma palavra. Por fim, a similaridade entre duas palavras é calculada pela similaridade do cosseno entre os vetores de cada palavra. Han et al. (2013) apresentam uma descrição detalhada do uso do modelo HAL com SVD para língua inglesa.

O modelo de similaridade semântica inferencialista, proposto por Pinheiro et al. (2014) e Pinheiro et al. (2010b) define a *Word Inferential Similarity Measure* a qual calcula a similaridade entre dois conceitos pela interseção entre o conjunto das pré-condições [ou pós-condições] de uso dos dois conceitos, aludindo a ideia de que quanto mais as circunstâncias [ou consequências] de uso de ambos os conceitos são similares, mas as inferências em que os mesmos podem participar são similares.

Han et al. (2013) propõem uma medida de similaridade entre palavras que agrega valor da base WordNet à medida LSA.

3.2.2 Estratégias de Alinhamento entre termos

A estratégia de alinhamento é necessária para definir quais termos de cada sentença serão comparados em termos de similaridade semântica. Considere os textos de entrada t_1 e t_2 com as seguintes sentenças $\{s_{11}, s_{12}, s_{13}\}$ e $\{s_{21}, s_{22}, s_{23}\}$, respectivamente. Na etapa anterior, os conjuntos T_{11} e T_{21} com os termos das sentenças s_{11} e s_{21} foram gerados. Propõe-se então uma função de alinhamento t-align(c) (Fórmula 1 que busca alinhar o termo c em T_{11} com um ou mais termos c' em T_{21}, de acordo com uma das seguintes estratégias:

1. *tokens* de mesma classe gramatical (*POS tag*) (p.ex. substantivo com substantivo, verbo com verbo, etc.);
2. *tokens* com mesma função sintática (p.ex. sujeito com sujeito, verbo principal com verbo principal, etc.);

3. *tokens* com maior valor de similaridade semântica entre palavras;

4. todos os *tokens* com todos;

Seguindo Han et al. (2013), a estratégia 3 alinha o termo c em T_{ij} com o termo c' em T_{lj}, que tiver maior valor de similaridade semântica $\theta(c, c')$ (Fórmula 1).

$$\text{t-align}(c) = \text{argmax}_{c' \in T_{lj}} \theta\left(c, c'\right). \quad (1)$$

A flexibilidade de adotar uma dentre várias estratégias de alinhamento permite adaptar o sistema STS a um domínio ou aplicação. No entanto, argumentamos que a estratégia 1 (que utiliza o critério de *POS tag*) e a estratégia 2 (que utiliza o critério de função sintática) são mais intuitivas e linguisticamente fundamentadas, embora mais complexas.

3.3 Similaridade Semântica Textual

Na última etapa do processo, o framework prevê duas abordagens para cálculo da STS—algoritmos de aprendizagem automática e/ou algoritmos simbólicos.

A abordagem por aprendizagem de máquina preconiza o uso de algoritmos supervisionados, tais como definidos por Chang & Lin (2011), Hall et al. (2009) e Pedregosa et al. (2011), com uso de características (*features*) sintáticas, lexicais e semânticas.

Na abordagem simbólica, a intuição básica de uma medida de similaridade semântica entre textos é que, quanto mais as sentenças dos textos são similares, mais os textos são similares. Da mesma forma, quanto mais os conceitos articulados nas sentenças são similares, mas similares as sentenças também serão. Neste sentido, a medida *SIMt* (Fórmula 4) define a similaridade entre dois textos de entrada t_1 e t_2 pela média da similaridade entre as sentenças s e s' que são mais similares. Ou seja, cada sentença s de t_1, é alinhada com a sentença s' de t_2 que lhe é mais similar.

A Fórmula 2 apresenta nossa função de alinhamento de sentenças s-align(s), a qual, para a sentença s de t_1 (ou t_2), retorna sua contraparte s' em t_2 (ou t_1), com maior valor da medida de similaridade entre sentenças *SIMs* (Fórmula 3).

$$\text{s-align}(s) = \text{argmax}_{s' \in t_i} \text{SIMs}\left(s, s'\right). \quad (2)$$

A Fórmula 3 define a medida de similaridade entre sentenças *SIMs* entre duas sentenças s_1 e s_2 pela média ponderada do somatório das similaridades entre seus termos alinhados.

$$\text{SIMs}\left(s_1, s_2\right) = \frac{\sum_{i=1}^{n} \sum_{j=1}^{q_i} \theta\left(c, c'\right) \times P_i}{\sum_{i=1}^{n} q_i \times P_i} \quad (3)$$

Onde:

- $\theta(c, c')$ é o valor da similaridade entre os *tokens* das sentenças s_1 e s_2, de acordo com o modelo de similaridade entre palavras definido na etapa anterior (seção 3.2.1);
- n é a quantidade de "tipos gramaticais" definidos na estratégia de alinhamento. Por exemplo, usando o critério de alinhamento por função sintática (estratégia 2), pode-se ter $n = 3$, conforme os seguintes tipos: SUJEITO, VERBAL PRINCIPAL e OBJETO;
- q_i é a quantidade de elementos em cada "tipo gramatical" i;
- P_i é o peso do "tipo gramatical" i, permitindo, por exemplo, que a similaridade entre verbos tenha um peso maior que a similaridade entre objetos diretos.

Finalmente, a Fórmula 4 calcula a similaridade semântica entre dois textos de entrada t_1 e t_2, com p e k sentenças, respectivamente.

$$\begin{aligned} \text{SIMt}(t_1, t_2) &= \frac{\sum_{s \in t_1} \text{SIMs}(s, \text{s-align}(s))}{2p} \\ &+ \frac{\sum_{s \in t_2} \text{SIMs}(s, \text{s-align}(s))}{2k} \end{aligned} \quad (4)$$

Pinheiro et al. (2014) apresentam um exemplo ilustrativo de uso das fórmulas acima.

4 Sistemas STS para ASSIN

O framework FlexSTS foi usado para instanciar três sistemas para STS na língua portuguesa, cujos resultados foram submetidos à avaliação no Workshop de Avaliação de Similaridade Semântica e Inferência Textual (ASSIN), realizado durante a conferência PROPOR 2016. A seguir serão explanadas a configuração de cada sistema e do sistema *baseline*. Ao final, os resultados e uma discussão dos mesmos serão apresentados.

Importante aqui salientar a flexibilidade do *framework* FlexSTS onde podem ser mesclados diversos componentes para instanciar ou criar sistemas de STS. Basicamente são selecionados componentes para cada etapa do processo:

Análise Morfológica, Similaridade entre Palavras, e Similaridade entre Textos. As tabelas 1, 2 e 3 apresentadas nas subseções a seguir detalham os componentes utilizados em cada sistema. A escolha dos componentes visou combinar abordagens simbólicas e estatísticas.

4.1 STS_MachineLearning

O sistema STS_MachineLearning aplicou uma abordagem híbrida para cálculo da STS — aprendizagem automática usando dois atributos (*features*) – similaridade entre palavras pelo coeficiente DICE e similaridade entre palavras pela WordNet. A configuração do sistema está descrita na Tabela 1.

Etapa	**Componente / Modelo**	**Ferramenta**
Análise Morfológica/Sintática	POS Tagger / Lematização	FreeLing
Similaridade Semântica de Palavras	Coeficiente DICE	Ver 4.1.1
	WordNet	Ver 4.1.1
Similaridade Semântica Textual	Aprendizagem Automática	Ridge Regression Model

Tabela 1: Configuração do sistema STS_MachineLearning.

4.1.1 Modelo de Aprendizagem de Máquina

No cálculo de STS foi usado o algoritmo *ridge regression model* (Pedregosa et al., 2011), um modelo de regressão com $\alpha = 1.0$ e um resolvedor automático que seleciona o peso de uma coleção dependendo do tipo de dado. Esses algoritmos foram usados por Sultan et al. (2015), campeão da tarefa de STS no SemEval 2015. O treinamento do algoritmo *ridge regression model* foi realizado com o *dataset* de treinamento disponibilizado na ASSIN. A seguir detalhamos os cálculos das duas *features* usadas para caracterizar o conjunto de exemplos.

Feature DICE

Esta *feature* representa a similaridade semântica textual entre os dois textos (exemplo) calculada pela Fórmula 4 usando a coeficiente DICE (Rohlf, 1992) como medida de similaridade entre palavras $\theta(c, c') = \text{DICE}(c, c')$. A Fórmula 5 define este cálculo.

$$\text{DICE}(c, c') = \begin{cases} 1 & \text{se} \begin{cases} isNum(c) \wedge isNum(c') \wedge c = c' \\ isCorrespondingPronoun(c, c') \\ diceCoefficient(c, c') > 2/3 \end{cases} \\ 0 & \text{caso contrário} \end{cases} \quad (5)$$

Onde,

- $isNum(c)$ retorna verdadeiro se c é um número;
- $isCorrespondingPronoun(c, c')$ verifica se os termos c e c' são pronomes correspondentes. Por exemplo, para os pronomes "*eu*" e "*me*" retorna verdadeiro;
- $diceCoefficient(c, c')$ calcula o coeficiente de Dice entre os termos c e c', conforme definido por Rohlf (1992).

Feature WNET

Esta *feature* representa a similaridade semântica textual entre os dois textos (exemplo) calculada pela Fórmula 4 usando conhecimento da WordNet para calcular a similaridade entre palavras, conforme Formula 6:

$$\text{WNET}'(c, c') = 0.5e^{\alpha D(c,c')} \quad (6)$$

Onde,

- $D(c, c')$ é uma função de distância entre os termos na base WordNet, calculado conforme segue:
  - 0, caso os termos pertençam ao mesmo conjunto de sinônimos (*synset*);
  - 1, nos seguintes casos: uma palavra é hiperonímia direta da outra; um adjetivo tem uma relação direta do tipo *similar to* com outro; uma palavra é uma forma derivacional da outra.
  - 2, nos seguintes casos: uma palavra é 2 *links* de hiperonímia indireta da outra; um adjetivo é 2 *links similar to* com outro; uma palavra é cabeça (*head*) do glossário da outra, ou sua hiperônima direta, ou uma das suas hipônimas diretas.
- α, parâmetro de normalização definido por Han et al. (2013) e fixado em 0,25.

A versão utilizada da WordNet foi a versão 3.0 em inglês e foi realizada a tradução dos corpus da

ASSIN (Português-Inglês) pelo Google Tradutor. A escolha desta solução deveu-se a dificuldades técnicas no uso da OpenWordNet.PT[4].

4.2 STS_LSA

O sistema STS_LSA aplicou somente a abordagem simbólica para cálculo da STS, usando o modelo LSA de similaridade entre palavras e a estratégia de alinhamento por termos com maior similaridade (estratégia 3). A configuração do sistema STS_LSA está descrita na Tabela 2

Etapa	**Componente / Modelo**	**Ferramenta**
Análise Morfológica/Sintática	POS Tagger / Lematização	FreeLing
Similaridade Semântica de Palavras	Modelo LSA (HAL+SVD)	Ver 4.2.1
	Estratégia de alinhamento	t-align$_3$ (fórmula 1)
Similaridade Semântica Textual	Altoritmo Matemático STS	Fórmulas 2, 3 e 4

Tabela 2: Configuração do sistema STS_LSA.

4.2.1 Modelo de Similaridade LSA

Foi usada a variação da técnica LSA chamada HAL (*Hyperspace Analog to Language*) (Burgess et al., 1998) que constrói a matriz de coocorrência termo-termo. Para a construção da msubatriz, foi usado o corpus CETENFolha[5] — um corpus de cerca de 24 milhões de palavras em Português-Brasileiro, com base nos textos do jornal Folha de S. Paulo que fazem parte do corpus do Núcleo Interinstitucional de Linguística Computacional (NILC), da USP/São Carlos.

Por questões de desempenho computacional, foram selecionados os 24000 termos que mais ocorrem no corpus, das classes abertas de palavras (substantivos, verbos, adjetivos e advérbios). Neste vocabulário não existem nomes próprios. A frequência de coocorrência entre os 24000 termos foi contada em uma janela de tamanho fixo que passa por todo o corpus. O tamanho de janela utilizado foi ±4, pois foi o que obteve melhor resultado por Han et al. (2013). Por fim, foi aplicada a estratégia de SVD (*Single Value Decomposition*) de Baglama & Reichel (2015), e selecionados os $k = 300$ maiores valores singulares. Assim, o tamanho do vetor que representa as palavras foi reduzido de 24000 para 300. A similaridade entre os termos foi calculada utilizando a função cosseno entre os vetores.

[4] http://wnpt.brlcloud.com/wn/

[5] http://www.linguateca.pt/cetenfolha/

4.3 STS_WORDNET_LSA

O sistema STS_WORDNET_LSA aplicou somente a abordagem simbólica para cálculo da STS, o modelo LSA de similaridade entre palavras e a estratégia de alinhamento por termos com maior similaridade (estratégia 3). Como conhecimento adicional, adicionou informação da WordNet no cálculo da similaridade LSA, a exemplo do trabalho de Han et al. (2013). A configuração do sistema STS_WORDNET_LSA está descrita na Tabela 3.

Etapa	**Componente / Modelo**	**Ferramenta**
Análise Morfológica/Sintática	POS Tagger / Lematização	FreeLing
Similaridade Semântica de Palavras	Modelo LSA (HAL+SVD)	Ver 4.2.1
	Estratégia de alinhamento	t-align$_3$ (fórmula 1)
	Base de Conhecimento / WordNet	Ver 4.3.1
Similaridade Semântica Textual	Altoritmo Matemático STS	Fórmulas 2, 3 e 4

Tabela 3: Configuração do sistema STS_WORDNT_LSA.

4.3.1 LSA + Conhecimento da WordNet

À medida de similaridade entre palavras $\theta(c, c') = \text{LSA}(c, c')$ (ver 3.2.1) foi adicionado conhecimento da base WordNet (Han et al., 2013). A Fórmula 7 apresenta este cálculo.

$$\text{WNET}(c, c') = \text{BASIC}(c, c') + \text{WNET}'(c, c') \quad (7)$$

$$\text{BASIC}(c, c') = \begin{cases} \theta(c, c') & \text{se } \theta \neq nulo \\ \text{DICE}(c, c') & \text{se } usaDice = \top \wedge \\ & (\theta = nulo \vee \theta(c, c') = 0) \\ 0 & \text{caso contário} \end{cases}$$

Onde,

- $\theta(c, c') = \text{LSA}(c, c')$ (ver 3.2.1);
- *usaDice* é um parâmetro que indica se, em caso valor $\theta(c, c')$ nulo ou zerado, deva-se usar o valor do coeficiente DICE;
- $\text{DICE}(c, c')$, conforme definido em Fórmula 5;
- $\text{WNET}'(c, c')$, conforme definido em Fórmula 6.

4.4 STS_Baseline

O sistema *STS_Baseline* foi usado neste trabalho apenas como referência inicial de avaliação, visto que, antes da ASSIN, inexistia estado da arte para STS em língua portuguesa. Nossa proposta foi utilizar o coeficiente de similaridade DICE (conforme definido em 3.1), como sistema *baseline* para a tarefa de STS.

4.5 Resultados e Discussão

A tabela 4 apresenta os resultados da medida de correlação de *Pearson* dos três sistemas STS (*runs*), enviados para ASSIN, após execução no *dataset* de teste para Português-Brasileiro (PT-BR) e Português-Portugal (PT-PT). Nosso melhor sistema foi o STS-MachineLearning em ambos os *datasets*. Na última linha da Tabela 4, apresentamos os resultados do sistema *baseline*, que obteve melhor desempenho que qualquer um dos sistemas avaliados para PT-PT.

Sistema	**PT-BR**	**PT-PT**
STS_MachineLearning	**0,62**	**0,64**
STS_LSA	0,56	0,59
STS_WNET_LSA	0,61	0,63
STS_Baseline	0,60	**0,69**

Tabela 4: Resultados dos sistemas STS desenvolvidos a partir do *framework* FlexSTS.

A seguir elencamos duas dificuldades importantes enfrentadas na construção dos sistemas de STS submetidos à ASSIN:

- No sistema STS_LSA, a matriz de coocorrência termo-termo gerada era muito esparsa, implicando em pouca relevância do cálculo da similaridade pela LSA. Atribui-se como causa o tamanho do corpus e tamanho dos textos do corpus;
- O uso da versão em Inglês da WordNet com a necessidade de solução de tradução Português-Inglês dos corpus ASSIN pode ter prejudicado o desempenho dos sistemas que utilizam esta base.

O uso do sistema *baseline* pelo coeficiente DICE permitiu constatar que uma medida simples de similaridade sintática obteve resultado significativo em relação aos corpus PT-BR (0,60) e PT-PT (0,69). Em apenas 211 casos do corpus *Gold Standard* ASSIN, o valor absoluto da diferença entre o valor da similaridade DICE e o valor GOLD foi superior a 2 ($|DICE - GOLD| > 2$). No demais casos (1935), estes valores são muito próximos. Portanto, conclui-se que os corpus ASSIN possuem uma similaridade lexical alta, dificultando a influência de conhecimento semântico à tarefa de STS.

Analisando alguns casos em que o sistema STS_MachineLearning obteve melhor resultado comparado com a solução baseline (DICE), identificamos que conhecimento semântico agregou valor à tarefa. Por exemplo, para o par de texto t_1 e t_2 na Figura 2, o sistema STS_MachineLearning apresentou valor de similaridade mais correlato ao valor GOLD, pois encontrou valor de similaridade entre as palavras "*intervalo*" e "*tempo*".

> t_1 = "O time treinado por Rafa Benítez assumiu uma postura covarde em o segundo **tempo** e apenas se defendeu"
> t_2 = "O time voltou de o **intervalo** com uma postura covarde e passou a apenas se defender"

Figura 2: Exemplo de textos com uso de conhecimento da WordNet.

5 Conclusão

Neste trabalho apresentamos a proposta do *framework* FlexSTS, o qual define diversos componentes a serem conectados para o desenvolvimento de sistemas de STS, agregando modelos e medidas de similaridade, *toolkits* e algoritmos do estado da arte, em cada etapa do processo de STS.

FlexSTS foi instanciado em três sistemas:

1. STS_MachineLearning: abordagem híbrida para cálculo da STS com aprendizagem automática usando dois atributos (*features*) — similaridade entre palavras pelo coeficiente DICE e similaridade entre palavras pela WordNet;

2. STS_LSA: abordagem simbólica que usa basicamente o modelo de similaridade de palavras da *Latent Semantic Analysis* (LSA);

3. STS_WORDNET_LSA: uma abordagem também simbólica que agrega conhecimento da WordNet à similaridade pela LSA.

Os sistemas foram testados nos *datasets* de teste disponíveis na ASSIN para Português-Brasileiro (PT-BR) e Português-Portugal (PT-PT). Nosso melhor sistema foi o STS-MachineLearning com resultado para o PT-PT de 0,64 (correlação de Pearson). Os principais problemas foram a esparsidade da matriz de coocorrência termo-termo construída a partir do corpus CETEMFolha e o uso da WordNet em inglês. Um resultado importante foi o desempenho do sistema *baseline* pelo coeficiente de DICE, que obteve 0,69 para o *corpus* PT-PT, indiciando que os corpus possuem alta similaridade lexical.

A análise dos resultados, dos problemas enfrentados e de erros do sistema indicam os seguintes trabalhos futuros: criação de mais cenários de testes com diversificação de algoritmos de *machine learning* e novas *features*; construção de nova matriz LSA a partir de um corpus mais robusto na língua portuguesa; agregação de conhecimento da Wikipedia e InferenceNet.

Referências

Agirre, Eneko, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre & Weiwei Guo. 2013. *SEM 2013 shared task: Semantic textual similarity. Em *Second Joint Conference on Lexical and Computational Semantics (*SEM)*, 32–43.

Albuquerque, Adriano, Vládia Pinheiro & Thiago Leite. 2012. Reuse of experiences applied to requirements engineering: An approach based on natural language processing. Em *Proceedings of the 24th International Conference on Software Engineering & Knowledge Engineering (SEKE'2012)*, 574–577.

Baglama, Jim & Lothar Reichel. 2015. *irlba: Fast truncated svd, pca and symmetric eigen decomposition for large dense and sparse matrices. r package version 2.0.0.*

Baldridge, Jason. 2005. The OpenNLP project. http://opennlp.apache.org.

Burgess, Curt, Kay Livesay & Kevin Lund. 1998. Explorations in context space: Words, sentences, discourse. *Discourse Processes* 25(2–3). 211–257.

Chang, Chih-Chung & Chih-Jen Lin. 2011. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* 2(3). 27:1–27:27.

Ganitkevitch, Juri, Benjamin Van Durme & Chris Callison-Burch. 2013. PPDB: The paraphrase database. Em *Proceedings of NAACL-HLT*, 758–764.

Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann & Ian H. Witten. 2009. The weka data mining software: An update. *SIGKDD Explorations Newsletter* 11(1). 10–18.

Han, Lushan, Abhay L. Kashyap, Tim Finin, James Mayfield & Johnathan Weese. 2013. UMBC_EBIQUITY-CORE: Semantic Textual Similarity Systems. Em *Proceedings of the Second Joint Conference on Lexical and Computational Semantics*, 44–52. ACL.

Harris, Zellig. 1968. *Mathematical structures of language*. Wiley.

Landauer, Thomas & Susan Dumais. 1997. A solution to Plato's problem: The Latent Semantic Analysis theory of the acquisition, induction, and representation of knowledge. *Psychological Review* 104(2). 211–240.

Lin, Chin-Yew & Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. Em *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1* NAACL '03, 71–78.

Liu, Hugo & Push Singh. 2004. Conceptnet — a practical commonsense reasoning tool-kit. *BT Technology Journal* 22(4). 211–226.

Miller, George A. 1995. Wordnet: A lexical database for english. *Communications of the ACM* 38. 39–41.

Milne, David & Ian H. Witten. 2008. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. Em *Proceeding of AAAI Workshop on Wikipedia and Artificial Intelligence: an Evolving Synergy*, 25–30.

Padró, Lluís & Evgeny Stanilovsky. 2012. Freeling 3.0: Towards wider multilinguality. Em *Language Resources Evaluation Conference*, 2473–2479.

Pedregosa, Fabian, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake

Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot & Édouard Duchesnay. 2011. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research* 12. 2825–2830.

Pinheiro, Vládia, Vasco Furtado & Adriano Albuquerque. 2014. Semantic textual similarity of Portuguese-language texts: An approach based on the semantic inferentialism model. Em Jorge Baptista, Nuno Mamede, Sara Candeias, Ivandré Paraboni, Thiago A. S. Pardo & Maria das Graças Volpe Nunes (eds.), *Computational Processing of the Portuguese Language: 11th International Conference*, 183–188.

Pinheiro, Vládia, Vasco Furtado, Lívio Melo Freire & Caio Ferreira. 2012. Knowledge-intensive word disambiguation via commonsense and wikipedia. Em *Proceedings of the 21st Brazilian Conference on Advances in Artificial Intelligence* SBIA'12, 182–191. Springer-Verlag.

Pinheiro, Vladia, Tarcisio Pequeno, Vasco Furtado & Wellington Franco. 2010a. InferenceNet.Br: Expression of inferentialist semantic content of the portuguese language. Em Thiago Alexandre Salgueiro Pardo, António Branco, Aldebaro Klautau, Renata Vieira & Vera Lúcia Strube de Lima (eds.), *Computational Processing of the Portuguese Language: 9th International Conference*, 90–99.

Pinheiro, Vládia, Tarcisio Pequeno & Vasco Furtado. 2010b. Um analisador semântico inferencialista de sentenças em linguagem natural. *Linguamática* 2(1). 111–130.

Rohlf, F. James. 1992. *Numerical taxonomy and multivariate analysis system*. Department of Ecology and Evolution, State University of New York.

Sultan, Md Arafat, Steven Bethard & Tamara Sumner. 2015. Dls@cu: Sentence similarity from word alignment and semantic vector composition. Em *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, 148–153.

Toutanova, Kristina, Dan Klein, Christopher D. Manning & Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. Em *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1* NAACL'03, 173–180.

Zhao, Jiang, Tiantian Zhu & Man Lan. 2014. ECNU: one stone two birds: Ensemble of heterogenous measures for semantic relatedness and textual entailment. Em *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval-COLING 2014*, 271–277.