Universidade do Minho

Universidade de Vigo

linguaMATICA

Volume 14, Número 1 (2022)

ISSN: 1647-0818

linguaMATICA

Volume 14, Número 1 – 2022

LinguaMÁTICA

ISSN: 1647–0818

Conteúdo

Artigos de Investigação

Editorial

Depois de treze anos, e devido a diversas razões, tornou-se imperativo renovar a equipa editorial da Linguamática. Embora os editores fundadores se mantenham ativos, foram convidados dois investigadores, revisores e autores da Linguamática, a assumir o papel de Editores Executivos.

Deste modo, foram convidados o Hugo Gonçalo Oliveira, da Universidade de Coimbra, e o Marcos Garcia, da Universidade de Santiago de Compostela, que com ânimo aceitaram o desafio.

Temos a certeza que irão primar pela qualidade, tal como têm feito no trabalho de investigação que fazem, e que deste modo continuarão com o objetivo de manter a Linguamática como uma referência para o processamento da linguagem natural.

Alberto Simões
Xavier Gómez Guinovart
José João Almeida

A Linguamática nasceu há mais de 13 anos, apostando numa política de publicação completamente aberta e promovendo a investigação computacional das e nas línguas ibéricas. O tempo confirmou que aquela arriscada aposta foi não apenas bem sucedida, mas também em certo modo visionária, sendo hoje tanto a publicação em aberto como a perspectiva multilíngue no processamento computacional das línguas tendências dominantes na investigação nesta área.

Também nós olhamos há muito para a Linguamática como uma referência. É por isso com enorme prazer, mas também com grande sentido de responsabilidade, que aceitamos a proposta da anterior equipa editorial da Linguamática para assumirmos o papel de editores da revista. Queremos assim continuar a promover a investigação e a publicação de trabalhos de carácter científico sobre as línguas ibéricas, fazendo pontes entre os diferentes povos da Península Ibérica e toda a comunidade internacional interessada nestas línguas.

Queremos agradecer tanto à equipa editorial que finaliza a sua etapa, cujo trabalho realizado facilita a nossa missão, como à comissão científica e editores convidados que ao longo deste tempo colaboraram na leitura e revisão dos trabalhos submetidos. Tudo faremos para manter o nível a que a Linguamática nos habituou.

Hugo Gonçalo Oliveira
Marcos Garcia

Comissão Científica

Alberto Álvarez Lugrís,
Universidade de Vigo

Alberto Simões,
Instituto Politécnico do Cávado e Ave

Aline Villavicencio,
Universidade Federal do
Rio Grande do Sul

Álvaro Iriarte Sanroman,
Universidade do Minho

Anselmo Peñas,
Universidad Nacional de
Educación a Distancia

Antoni Oliver González,
Universitat Oberta de Catalunya

Antonio Moreno Sandoval,
Universidad Autónoma de Madrid

António Teixeira,
Universidade de Aveiro

Arkaitz Zubiaga,
Dublin Institute of Technology

Bruno Martins,
Instituto Superior Técnico

Carmen García Mateo,
Universidade de Vigo

Diana Santos,
Linguateca/Universidade de Oslo

Fernando Batista,
Instituto Universitário de Lisboa

Ferran Pla,
Universitat Politècnica de València

Gael Harry Dias,
Université de Caen Basse-Normandie

Gerardo Sierra,
Universidad Nacional
Autónoma de México

German Rigau,
Euskal Herriko Unibertsitatea

Helena de Medeiros Caseli,
Universidade Federal de São Carlos

Horacio Saggion,
University of Sheffield

Hugo Gonçalo Oliveira,
Universidade de Coimbra

Irene Castellón Masalles,
Universitat de Barcelona

Iria da Cunha,
Universidad Nacional de
Educación a Distancia

Itziar Gonzalez-Dios,
Euskal Herriko Unibertsitatea

Joaquim Llisterri,
Universitat Autònoma de Barcelona

Jorge Baptista,
Universidade do Algarve

José João Almeida,
Universidade do Minho

José Paulo Leal,
Universidade do Porto

Juan-Manuel Torres-Moreno,
Université d´Avignon et
des Pays du Vaucluse

Kepa Sarasola,
Euskal Herriko Unibertsitatea

Laura Plaza,
Complutense University of Madrid

Liliana Ferreira,
Fraunhofer Portugal AICOS & FEUP

Lluís Padró,
Universitat Politècnica de Catalunya

Luís Morgado da Costa,
Nanyang Technological University

Manex Agirrezabal,
University of Copenhagen

Marcos Garcia,
Universidade de Santiago de Compostela

María Inés Torres,
Euskal Herriko Unibertsitatea

Mário Rodrigues,
Universidade de Aveiro

Mercè Lorente Casafont,
Universitat Pompeu Fabra

Miguel Solla Portela,
Universidade de Vigo

Mikel Forcada,
Universitat d'Alacant

Pablo Gamallo Otero,
Universidade de Santiago de
Compostela

Patrícia Cunha França,
Universidade do Minho

Patricia Martin Rodilla
Universidade de Santiago de
Compostela

Ricardo Rodrigues
Instituto Politécnico de Coimbra

Rogelio Nazar
Pontificia Universidad Católica de Valparaíso

Rui Pedro Marques,
Universidade de Lisboa

Sebastião Pais,
Universidade da Beira Interior

Susana Afonso Cavadas,
University of Exeter

Xavier Gómez Guinovart,
Universidade de Vigo

Artigos de Investigação

Propuesta recibida en noviembre 2020 y aceptada para publicación en abril 2022.

La #felicidad en Twitter: ¿qué representa realmente?

#happiness in Twitter: What does it really represent?

Gemma Bel-Enguix
Instituto de Ingeniería
Universidad Nacional Autónoma de México

Helena Gómez-Adorno
Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas
Universidad Nacional Autónoma de México

Karla Mendoza Grageda
Facultad de Ciencias
Universidad Nacional Autónoma de México

Grigori Sidorov
Centro de Investigación en Computación
Instituto Politécnico Nacional

Juan Vásquez
Posgrado en Ciencia e Ingeniería de la Computación
Universidad Nacional Autónoma de México

Resumen

Existe un gran número de trabajos que tienen por objeto la clasificación de diversos tipos de documentos, desde textos literarios hasta interacciones informales en redes sociales como Twitter, de acuerdo a los sentimientos que pretenden evocar. Se pueden realizar clasificaciones muy variadas con base en los sentimientos que el autor considere. El objetivo de este artículo es clasificar una recopilación de tuits en diferentes contextos en los que la palabra 'feliz' o 'felicidad' se pueden emplear; por ejemplo publicidad, felicitaciones o como un simple sarcasmo. Para esto se hará uso de sistemas de aprendizaje supervisado y se emplearán varios métodos de procesamiento de lenguaje natural como tokenización, identificación de palabras funcionales y n-gramas.

Palabras clave

Feliz, Felicidad, Tuit, Aprendizaje Automático

Abstract

There are a number of works that deal with the classification of feelings evoked by various types of documents, from literaty texts to social networks like Twitter. Various classifications can be performed based on the sentiments considered by the author. The goal of this article is to classify a collection of tweets in different contexts in which the word 'happy' or 'happiness' can be used, for example advertising, congratulations or as a simple sarcasm. This will be done using supervised learning systems. Several natural language processing methods such as tokenization, functional word identification, and n-grams will be employed.

Keywords

Happy, Happiness, Tweet, Machine Learning

1. Introducción

Las redes sociales se han convertido en el medio de comunicación por excelencia en el siglo XXI. Al contrario de los denominados "mass media", que proliferaron y consiguieron un gran poder durante el siglo XX, las redes sociales tienen un carácter bidireccional, o incluso multidireccional. Esto significa que cada usuario es a la vez hablante y oyente, generador y receptor del contenido comunicativo. El hecho de que todos sean comunicadores contrasta con una de las principales características de los medios tradicionales: la exclusividad en la información y la difusión de la noticia. En este nuevo escenario, las redes sociales están más orientadas a la comunicación de actividades personales, sentimientos y opiniones, eso sí, en un canal que los difunde como si fueran noticias. En este contexto, la expresión de la emoción ha tomado un lugar preponderante en las plataformas de comunicación social.

Por otra parte, la generación de grandes cantidades de datos textuales y la aparición de técnicas de análisis computarizadas ha desembocado en un gran desarrollo de la minería de opinión y análisis de sentimientos (Turney, 2002). Los pri-

DOI: 10.21814/lm.14.1.342

meros trabajos en el área tenían como objetivo la clasificación de textos bajo las categorías de polaridad positivo/negativo. En diferentes contextos, esta polaridad se puede interpretar como buenas noticias vs. malas noticias (Koppel & Shtrimberg, 2004), me gusta vs. no me gusta (Kim & Hovy, 2006), estoy de acuerdo vs. no estoy de acuerdo (Bansal et al., 2008; Wojatzki et al., 2018), apoyo político vs. no apoyo (Thomas et al., 2006), o probablemente gane vs. probablemente no gane (Kim & Hovy, 2007) en el ámbito electoral.

Al principio, los trabajos se centraban en el análisis de reseñas y recomendaciones sobre diversos productos, desde películas a hoteles. Pero la gran penetración de las redes sociales en la vida cotidiana, ha acabado poniendo estas formas de comunicación en el centro de interés del *big data* por lo que respecta al lenguaje natural. Uno de los primeros trabajos que traslada el tema de clasificación de polaridad a análisis de grandes cantidades de datos extraídos de redes sociales es el de Go et al. (2009).

A veces, los escritos, sobre todo los originados en internet, no están orientados exclusivamente a dar la opinión sobre un tema en concreto, o a expresarla de una forma binaria. Por esto muchos autores optaron por identificar elementos de subjetividad y emociones en los textos (Banea et al., 2011; Morency et al., 2011; Mohammad & Kiritchenko, 2018). Durante la última década, muchos trabajos han seguido esta línea centrándose en el estudio de las redes sociales (Mohammad, 2012). El interés suscitado ha llevado a organizar competiciones en congresos internacionales (Mohammad et al., 2018; Naderi et al., 2018).

Aunque se han propuesto diversas formas de categorizar las emociones (Plutchik, 1980), existe un consenso general en considerar seis emociones básicas, siguiendo la clasificación de Ekman (1992; 1994). Estas son tristeza, miedo, enfado, disgusto, sorpresa y felicidad.

En general, dentro del marco descrito por Ekman, las emociones negativas han recibido más atención por parte de la comunidad científica. En cambio, este artículo va enfocado al estudio de la felicidad en Twitter. Las bases para esta investigación se encuentran en (Sidorov et al., 2016), donde se explica la metodología de la recopilación de tuits, basada en hashtags. Los mismos autores aplican estos criterios al filtrado y análisis de micro-mensajes con emociones negativas (Camacho Vázquez et al., 2018). En este trabajo se parte de un corpus que selecciona tuits con hashtags relacionados con la felicidad, cuyo léxico se incluye también dentro de este campo semántico. Sin embargo, nos preguntamos si el uso de una terminología relacionada con la felicidad implica que los tuits analizados transmiten efectivamente esta emoción o bien tienen otras connotaciones diferentes. Para verificar si efectivamente tienen otros significados, se propone realizar una detección manual y posteriormente implementar un sistema de aprendizaje que sea capaz de distinguir la verdadera emoción más allá del léxico y hashtags utilizados.

Para llevar a cabo esta investigación, el artículo está organizado de la siguiente forma. La sección 2 revisa el trabajo que se ha hecho hasta la fecha en esta misma línea de estudio. En la sección 3 se explica el proceso de compilación y etiquetado del corpus. Además, se detalla la metodología para la clasificación de tuits y se da cuenta de un primer acercamiento al léxico. La sección 4 reseña los principales de pasos de preprocesamiento textual que se han llevado a cabo. La sección 5 muestra los resultados obtenidos para cada uno de los experimentos. El artículo se cierra con las conclusiones (sección 6), donde se anotan también algunas líneas de trabajo futuro.

2. Trabajo Relacionado

El presente artículo aborda el estudio de la 'felicidad' en un corpus de tuits en español, proponiendo un sistema de aprendizaje automático que ayude a distinguir el verdadero significado de los micro-textos. Los artículos que sirven como referencia, o bien trabajan con la detección y análisis de las emociones, principalmente positivas, encontradas en diferentes corpus, o bien aplican técnicas de aprendizaje automático sobre colecciones textuales en español.

Aunque la mayor parte de la investigación en el área se realiza en inglés, existen interesantes contribuciones en español. Algunos artículos proponen algoritmos para la clasificación de polaridad (positivo-negativo-neutro) en diversos documentos. Por ejemplo, Vilares et al. (2013) clasifican textos subjetivos como positivos o negativos, basándose en diccionarios semánticos y en la estructura semántica de las oraciones.

Por otro lado, Gruzd et al. (2011) realizan un análisis sobre un conjunto de tuits recopilados desde el primer día de los juegos olímpicos hasta unos días antes del final donde, después de clasificarlos en positivo, negativo, neutro y ambos, llegan a la conclusión de que aquellos señalados como positivos son, en cantidad, tres veces más que aquellos señalados como negativos.

El trabajo de Mogilner et al. (2011), aunque en inglés, sigue una línea de interés muy cercana a la nuestra. Los autores recopilan blogs de cuyos

autores conocen la edad, derivados de las búsquedas de las frases 'yo siento' o 'me siento'. De esta recopilación filtran los que completan la frase de búsqueda con la palabra 'felicidad'. Después realizan un análisis sobre las palabras coocurrentes de esta frase distinguiendo dos categorías distintas *'excited happiness' y 'peaceful happiness'* de las cuales concluyen que los autores más jóvenes expresan una *excited happiness* mientras que los adultos están más cercanos a una *peaceful happiness*.

Kumar et al. (2015) trabajan con una recopilación de tuits y consideran cinco emociones derivadas de la clasificación de Ekman, de donde eliminan la sorpresa: felicidad, tristeza, disgusto, miedo e ira. Los autores evalúan adjetivos asignándoles un valor en cada uno de los sentimientos. Además consideran los adverbios y algunos verbos que modifican el adjetivo. De esta forma se calcula el valor del sentimiento para cada uno de los tuits recopilados.

Los ejemplos anteriores utilizan métodos usuales de sistemas de aprendizaje. Recientemente se ha procurado resolver problemas de clasificación con métodos basados en redes neuronales. Bakhtiyar et al. (2019) enfrentan el problema de distinguir entre 'felicidad del autor' y 'felicidad social'. Para ello proponen una transferencia inductiva semi-supervisada de aprendizaje, en la cual los resultados obtenidos en la tarea actual dependen de la transferencia de conocimientos obtenidos en tareas anteriores. Su propuesta consta de tres pasos. El primero tiene como propósito pre-entrenar el modelo del lenguaje base del modelo AWD-LSTM (*Average-SGD (Stochastic Gradient Descent) Weight-Dropped Long Short Term Memory*). En el segundo paso se ajusta el lenguaje utilizando datos específicos de la tarea; además, en lugar de mantener la misma tasa de aprendizaje para las capas del modelo AWD-LSTM, esta se va modificando en cada ajuste de capa. En el último paso se adapta un *gradual unfreezing heuristic* que se encarga de que no todas las capas sean ajustadas al mismo tiempo y, en lugar de eso, empieza 'descongelando' la última de ellas para ajustarla y seguir con las que la preceden. Al final se obtiene una exactitud de 93 % para la detección de 'felicidad social' y una exactitud de 87 % para la detección de 'felicidad del autor'.

Con todo este marco de referencia, el presente trabajo está estrechamente relacionado con un primer análisis realizado por Camacho Vázquez et al. (2018) sobre tuits con carga negativa. Dicho artículo explica cómo se implementan pruebas sobre dos categorías, tuits detectados como negativos y otros catalogados como neutrales. Los resultados de estos experimentos han demostrado que el uso de del sistema de aprendizaje de distribución multinomial (MNB) con frecuecia de término — frecuencia inversa de documento (TF-IDF) obtiene los mejores resultados con una puntuación $F_1 = 0{,}962$ tomando en cuenta unibigramas de palabras y $F_1 = 0{,}960$ tomando unigramas de palabras. Tras llevar a cabo diversas pruebas con cuatro categorías diferentes de tuits negativos y una categoría de tuits neutrales se muestra cómo las mayores puntuaciones se obtuvieron al usar el sistema de aprendizaje MNB con valores de frecuencias de términos dando una puntuación de $F_1 = 0{,}664$ usando unibigramas y una puntuación de $F_1 = 0{,}663$ usando unibitrigramas. Estos resultados son mejores comparados con las mismas pruebas tomando las palabras lematizadas.

De dichos experimentos se concluye que la combinación de diversas características, como se puede ver con los unibigramas y los unibitrigramas con frecuencias de términos, mejora los resultados obtenidos en las pruebas con sistemas de aprendizaje. Además se prueba que si se limitan las características más frecuentes combinadas por categorías a 1000 elementos también se obtiene una mejora en los resultados.

Por otra parte modelos basados en BERT también se han implementado para la tarea de clasificación de tuits en español. Por ejemplo, González et al. (2021) han propuesto un modelo basado en BERT y pre-entrenado con tuits en español. El objetivo de su modelo es mejorar los resultados del estado del arte en cuanto a tareas de clasificación de tuits en español. Asimismo Zeng et al. (2021) han propuesto un sistema de aprendizaje profundo llamado Senti-BSAS, el cual, por medio de un mecanismo de atención junto con un cálculo de sentimientos basado en un análisis léxico, clasifica la felicidad de una oración en dos clases: una clase en la cual el autor es el motivo de dicha felicidad, y otra en la cual la felicidad es generada por agentes externos al mismo.

Chiorrini et al. (2021) aplicaron los modelos *Uncased BERT* y *Cased BERT* para el análisis de emociones de tuits. Este trabajo utiliza al *Tweet Emotion Intensity Dataset*, un corpus de tuits en inglés creado por Mohammad & Bravo-Marquez (2017). La *accuracy* de *BERT uncased* resultó en 0.89, mientras que la F_1 fue de 0.89. Respecto al modelo *BERT cased*, obtuvieron una *accuracy* de 0.90 y una F_1 de 0.91.

Respecto al tratamiento de tuits en español, Rosá & Chiruzzo (2021) clasificaron tuits en ocho y seis clases. Este trabajo utiliza una red LSTM

alimentada con características generadas por BETO; específicamente, la red toma el token CLF y el centroide de la representación de cada token generado por BETO. Después de obtener las características, clasificaron los tuits respecto a los sentimientos *anger, disgust, fear, joy, sadness, surprise, others*. Después de obtener las características, realizaron dos distintas clasificaciones de sus tuits. La primera clasificación fue respecto a las clases *anger, disgust, fear, joy, sadness, surprise, others*, mientras que la segunda fue considerando *anger, disgust, fear, joy, sadness, surprise*. La clase *others* fue considerada para aquellos tuits que recibieron diferentes emociones de acuerdo a diferentes anotadores, o a tuits neutrales. Los resultados del modelo de clasificación entrenado con el corpus que contiene la clase *others* fueron: precisión de 0.6860 y $F_1 = 0{,}6620$, mientras que para el modelo entrenado con el corpus sin la clase *others* se obtuvo una precisión de 0.7447 y una $F_1 = 0{,}7170$.

Estos recientes trabajos de clasificación de tuits utilizando arquitecturas basadas en BERT demuestran el potencial de dichas arquitecturas para obtener mejores resultados que las técnicas clásicas para clasificación.

Aunque muy cercanos conceptualmente a nuestra investigación, estos trabajos no pueden compararse directamente con el nuestro. Por una parte, la mayor parte de trabajos tratan de la subcategorización de la felicidad, pero lo hacen desde presupuestos binarios. Otros, como Rosá & Chiruzzo (2021) proponen una clasificación multiclase, pero consideran únicamente las emociones primarias, no las distinciones dentro de cada una de ellas, lo que dificulta una comparación directa.

3. Compilación y etiquetado del corpus

La técnica de recopilación del corpus usada en este trabajo se basa en (Sidorov et al., 2016), donde se analizan formas de resolver algunos de los problemas que se presentan dentro de la detección de sentimientos en microblogs.

En el referido trabajo se recopilaron tuits que contienen un hashtag relacionado con una de las emociones: alegría, ira, tristeza, asco, miedo o sorpresa. Es importante mencionar que se descartaron todos aquellos tuits que contuvieran hashtags de más de una emoción, como el siguiente:

```
Última noche en casa de mis padres! #feliz
pero también última noche que duermo
con mis niños #triste.
```

En el trabajo mencionado se hace un análisis de los hashtags que se usan en los tuits relacionados con la emoción 'alegría'. Son los siguientes: #felicidad, #feliz, #alegria, #felicidades #alegre y #contento(a), de los cuales #feliz y #felicidad resultaron ser los más comunes.

Para nuestro trabajo, se han recopilado 10048 tuits del 26 de agosto al 2 de diciembre de 2016. Todos tienen el hashtag #felicidad o #feliz, además de cualquier otro de los que han sido identificados dentro de la categoría que define esta emoción. Este ha sido el único criterio de recopilación de corpus. Es decir, todos los tuits emitidos durante este período con este hashtag han sido recogidos. No existe ningún otro filtro. El corpus recopilado está disponible[1].

Por otra parte, Sidorov et al. (2016) realizaron un estudio del uso de los hashtags relacionados con la felicidad. Se llegó a la conclusión de que la alegría es la emoción que se expresa más frecuentemente dentro de Twitter. Sin embargo, se encuentra en múltiples contextos que, curiosamente, no siempre denotan alegría o felicidad. Mediante un análisis semántico hemos clasificado los tuits con los hashtags #felicidad, #feliz y #alegría en cinco categorías:

1. Alegría (A). Tuits que reflejan realmente alegría o felicidad: `que #feliz me haces, me sacas una #sonrisa sin hacer nada.... #esperanza!`
2. Publicidad (P). Hace referencia a comerciales que ofrecen felicidad si compras un producto determinado: `para ser #feliz primero se piensa en #seguridad y al pensar en seguridad se piensa en ocq security group`
3. Felicitación (F). Felicitaciones de cumpleaños, otros eventos personales o fechas señaladas: `eres ejemplo vivo de gente luchadora que ama a su país y #feliz cumpleaños te mando un beso gigante desde méxi https://t.co/o1rlvxygud`
4. Consejo (C). Se trata, en general, de mensajes de autoayuda o reflexiones (pseudo-)filosóficas: `descubre como vivir en el presente para poder ser #feliz y tener # https://t.co/uzkuigvtht salud sin depender de na https://t.co/tuguwxfomm`
5. Sarcasmo o no_alegría (N). Tuits con doble sentido que no denotan alegría en realidad: `literalmente... estás viendo la #felicidad "lo que ves es una proteína de miosina arrastrando una endorfina a lo... https://t.co/kjtcjq6vzn`

[1] https://github.com/GIL-UNAM/TwitterHappiness

3.1. Etiquetado

Los 10048 tuits que conforman el corpus final fueron proporcionados a tres personas para que etiquetaran, según su criterio, cada uno de los tuits en las categorías de 'alegría' (A), 'publicidad' (P), 'felicitación' (F), 'consejo' (C) y 'sarcasmo o no_alegría' (N).

En un 86.2 % de los casos, al menos dos etiquetadores estuvieron de acuerdo en considerar el tuit dentro del mismo grupo. Aquellos mensajes en los que ninguno de los etiquetadores coincidieron se clasificaron en 'No agreement' (NA). Representaron un 13.81 % del conjunto del corpus.

En la Figura 1 se observa el porcentaje de tuits que fueron englobados en cada categoría. Las clases más numerosas son las de 'alegría', 'consejos' y 'no agreement'. El alto porcentaje de NA indica hasta qué punto este tipo de clasificación está influenciada por parámetros subjetivos.

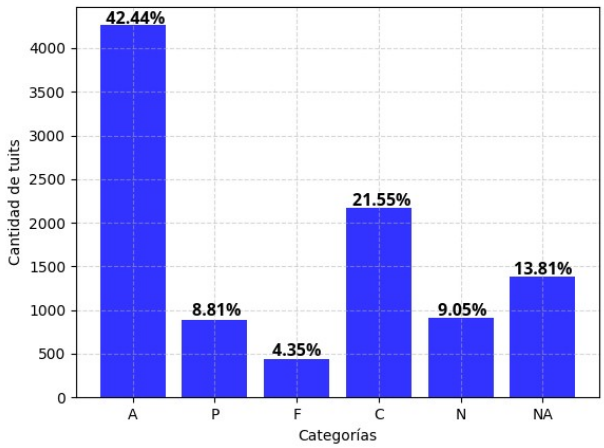

Figura 1: Gráfica de barras de la cantidad de tuits por categoría.

3.2. Análisis léxico

Una vez recopilado y etiquetado el corpus se ha realizado un análisis léxico comparando los resultados en cada una de las categorías. En primer lugar, después de extraer las palabras funcionales, se han seleccionado los 15 términos de contenido más frecuentes. Este número se ha establecido con el propósito de tener una muestra amplia de palabras pero suficientemente reducida para que pueda ser ilustrativa a la hora de su análisis. A continuación, se ha estudiado la frecuencia relativa de cada una de estas palabras en las diferentes categorías.

La Tabla 1 muestra los resultados obtenidos. A pesar de que *si* y *mas* se podrían considerar dentro de las palabras funcionales, en este caso se decidió conservarlas, pues podrían tener una estrecha relación con los sentimientos positivos. Hay que decir que la especial ortografía de la lengua tecleada no permite distinguir entre *si* y *sí*. Se puede observar que las palabras *feliz* y *felicidad* son las más frecuentes en todas las categorías y además están distribuidas equitativamente, con excepción de la palabra *felicidad* en la categoría F.

Palabra	**Categoría**					
	A	P	F	C	N	Total
feliz	7.71	6.42	10.71	6.51	6.99	7.33
felicidad	3.68	5.04	0.95	5.73	2.86	4.12
vida	0.80	0.58	0.76	2.00	0.36	1.01
hoy	1.01	0.75	1.43	0.43	0.61	0.82
día	0.72	0.62	1.80	0.66	0.60	0.77
si	0.40	0.54	0.32	1.26	0.69	0.66
mejor	0.59	0.49	0.55	0.62	0.31	0.55
gracias	0.88	0.25	0.76	0.13	0.14	0.53
amor	0.49	0.20	0.69	0.68	0.19	0.48
semana	0.42	0.36	0.21	0.27	0.34	0.45
siempre	0.39	0.12	0.53	0.72	0.19	0.44
mas	0.64	0.15	0.58	0.27	0.34	0.42
dios	0.40	0.00	1.02	0.43	0.12	0.40
cada	0.32	0.23	0.25	0.64	0.23	0.39
hace	0.38	0.15	0.21	0.35	0.28	0.32

Tabla 1: Frecuencia relativa de las primeras 15 palabras de mayor a menor aparición dentro de todo el corpus.

4. Procesamiento del corpus

Con el fin de poder trabajar con los textos de los tuits, estos se han sometido a un preprocesamiento que ha incluido: a) eliminación de los hiperenlaces, b) tokenizando, c) eliminación de los signos de puntuación y las palabras funcionales, d) extracción de raíces. Se han llevado a cabo experimentos combinando la existencia o no de los dos últimos procesos.

En este sentido, hay que mencionar que se intentaron otros métodos de preprocesamiento. Por ejemplo, se usó el lematizado en lugar de la extracción de raíces. Pero los resultados obtenidos fueron peores. Por ello, se optó por trabajar con esta última técnica.

En la Tabla 2 podemos ver una comparación entre un tuit antes y después de aplicar los pasos a, b y c del pre-procesamiento. Se observa que los hashtags se conservan, pues se espera que mejoren la posterior categorización, y se retira solamente el signo # de la palabra para estandarizar el vocabulario.

Tuit original	`#feliz #cumple a esa amiga hermosa que tengo la suerte de tener hace ya unos años. feliz de`
Tuit pre-procesado	`feliz cumple amiga hermosa suerte tener hace años feliz`

Tabla 2: Ejemplo de un tuit antes y después de pre-procesarse (pasos a, b, c).

Después del pre-procesamiento de los tuits podemos recabar la cantidad de palabras diferentes detectadas dentro de todo el corpus y la cantidad de palabras diferentes que resultan después de descartar las palabras funcionales, en la Tabla 3 se muestran dichas cantidades.

Cantidad de palabras distintas	19,357
Cantidad de palabras distintas descartando palabras funcionales	19,128

Tabla 3: Cantidad de palabras distintas dentro del corpus.

5. Clasificación automática de tipos de felicidad

Una vez compilado, etiquetado y pre-procesado el corpus, entrenamos algoritmos de aprendizaje para obtener modelos de clasificación de tipos de tuits con etiqueta de #felicidad. Los algoritmos utilizados son *Naive Bayes* (NB), *Logistic Regression* (LR), *Random Forest* (RF) y *Support Vector Machines* (SVM).

Para la representación vectorial de los tuits utilizamos n-gramas de palabras (con n de 1 a 6). Los n-gramas se forman después de realizar las operaciones de pre-procesamiento, es decir, cuando en el pre-procesamiento se eliminan las palabras funcionales éstas ya no forman parte de los n-gramas. Como podemos observar en las tablas 4 y 5, los n-gramas con $n = 1$ no parecen ser muy representativos ya que obtienen los resultados mas bajos cuando se realiza la evaluación del modelo de clasificación. Seguramente esto es a causa de que los vocabularios de cada categoría son bastante similares. Por ello se optó por incluir también secuencias más largas.

Evaluamos el impacto de las técnicas de pre-procesamiento descritas anteriormente en el rendimiento de los modelos de clasificación. Según lo que se aprecia en la Tabla 4, los mejores resultados se obtienen conservando las palabras funcionales. Esto resulta contraintuitivo, ya que estas palabras no aportan información semántica y, en principio, su eliminación debería aumentar el rendimiento de los modelos. Por el contrario, se espera que la extracción de raíces de las palabras disminuya el rendimiento de los modelos ya que, por ejemplo, 'felicidad' y 'felicidades' tienen la misma raíz. Si suponemos que 'felicidades' puede ser una palabra única dentro de la categoría 'F', al extraer la raíz dejará de serlo. En cambio, se obtienen mejores resultados cuando se realiza la extracción de raíces.

Por último, realizamos experimentos con dos esquemas de pesado frecuentemente utilizados en la literatura existente en el área: la frecuencia del término (TF) y la frecuencia del término por la frecuencia inversa de documento (TF-IDF). En este experimento se espera que, como la medida TF-IDF considera la especificidad de los términos dentro del corpus, mejore los resultados de la clasificación de los tuits.

Para evaluar la generalización de los modelos de clasificación utilizamos un método de validación cruzada con el fin de asegurar que los resultados fueran independientes de la partición de corpus en los conjuntos de prueba y entrenamiento. En particular, utilizamos un método de validación cruzada estratificada por capas que permite preservar la proporción de muestras para cada clase pues, como se puede observar en las cantidades de tuits por categoría y en la gráfica de barras, no se tiene instancias balanceados por clase.

Se realizaron experimentos con validación cruzada para 3, 5 y 10 capas y se obtuvo un promedio de exactitud para cada uno de los algoritmos de aprendizaje implementados. Después se calculó un promedio de rendimiento de todos los clasificadores por tipo de pre-procesamiento y esquemas de pesado. Una vez teniendo estos promedios se procedió a identificar el tipo de pre-procesamiento y esquema de pesado más adecuado para cada conjunto de características. Los resultados del proceso descrito se muestran en la Tabla 4 donde se reporta el promedio del rendimiento de los 4 clasificadores por conjunto de características. Como resultado podemos observar que para cada uno de los conjuntos de características, los mejores porcentajes se obtienen cuando se consideran las palabras funcionales, se aplica el proceso de extracción de raíces y se utiliza el esquema de pesado TF-IDF. Nótese que este resultado es contrario a nuestra hipótesis de trabajo, que señalaba que el proceso de extracción de raíces tendría previsiblemente un impacto negativo en los resultados.

Configuraciones			Conjunto de características					
Con Palabras Funcionales	Raíz	TF-IDF	1-grama	1-2-grama	1-3-grama	1-4-grama	1-5-grama	1-6-grama
✓	✓	✓	**70.44 %**	**71.18 %**	**71.09 %**	**71.15 %**	**71.08 %**	**71.12 %**
✓	✓	✗	70.39 %	70.36 %	70.38 %	70.53 %	70.43 %	70.47 %
✓	✗	✓	70.17 %	70.61 %	70.59 %	70.56 %	70.66 %	70.57 %
✗	✓	✓	68.91 %	69.81 %	69.77 %	69.79 %	69.92 %	69.84 %
✗	✓	✗	68.47 %	69.57 %	69.44 %	69.59 %	69.63 %	69.61 %
✗	✗	✓	68.48 %	69.28 %	69.19 %	69.11 %	69.14 %	69.11 %
✓	✗	✗	70.17 %	70.08 %	70.15 %	70.10 %	70.10 %	70.10 %
✗	✗	✗	67.64 %	68.18 %	68.01 %	68.07 %	68.07 %	68.00 %

Tabla 4: Promedios de exactitud de los clasificadores por conjuntos de características (n-gramas de palabras) y diferentes configuraciones.

Características	NB	LR	RF	SVM	Promedio
1-grama	69.91 %	69.54 %	70.12 %	72.18 %	70.44 %
1-2-grama	**70.91 %**	70.20 %	70.91 %	72.69 %	**71.18 %**
1-3-grama	70.69 %	70.15 %	70.96 %	72.57 %	71.09 %
1-4-grama	70.63 %	**70.39 %**	70.85 %	**72.74 %**	**71.15 %**
1-5-grama	70.70 %	70.11 %	**70.97 %**	72.53 %	71.08 %
1-6-grama	70.64 %	70.28 %	70.95 %	72.59 %	71.12 %
	70.58 %	70.11 %	70.79 %	**72.55 %**	

Tabla 5: Promedios de exactitud de clasificadores por conjuntos de características, utilizando las siguientes configuraciones: con palabras funcionales, con extracción de raíces y TF-IDF.

Para obtener el mejor modelo de clasificación, buscamos el algoritmo de clasificación que funcione mejor con características específicas. Se calculó un promedio de los porcentajes de validación cruzada de cada algoritmo de aprendizaje con los pre-procesamientos. Los resultados se pueden observar en la Tabla 5, donde señalamos los mejores porcentajes de exactitud para cada uno de los sistemas de aprendizaje los cuales son 70.91 % para NB, 70.39 % para LR, 70.97 % para RF y 72.74 % para SVM. Aquí se puede observar que diferentes conjuntos de características funcionan mejor con diferentes clasificadores, sin embargo en promedio, los 2-gramas y los 4-gramas tienen un mejor rendimiento. En cuanto al desempeño promedio de los clasificadores, se puede observar que las SVM obtienen mejores predicciones que el resto en términos de exactitud.

Como hipótesis adicional se consideraron n-gramas de caracteres y se realizaron los mismos experimentos obteniendo la Tabla 6 donde se reporta el promedio del rendimiento de los 4 clasificadores por conjuntos de características. Podemos observar que se obtienen promedios bajos en comparación con la Tabla 4. Por lo anterior, concluimos que para este problema las características extraídas a nivel de palabra funcionan mejor.

Considerando que el modelo SVM fue el algoritmo con mejores promedios de exactitud, en la Tabla 7 mostramos su reporte de clasificación donde podemos observar que la clase en la que se tuvo una mayor precisión y exhaustividad fue la clase de 'alegría' obteniendo un porcentaje del 81 % en predicciones correctas, mientras que la clase 'no_alegría' obtuvo el promedio más bajo en predicciones correctas lo que podríamos asociar con una deficiencia en la detección de sarcasmo dentro de los tuits. Finalmente, el promedio de precisión del algoritmo entre todas las clases fue de un 75 % mientras que su promedio de exhaustividad fue de un 59 %, esto nos da un porcentaje de predicciones correctas del 63 % hechas por el modelo SVM de acuerdo a la medida F_1 y una exactitud del 74 % en sus predicciones.

Adicionalmente, se muestran las características que han resultado más relevantes para la clasificación en las clases A (Figura 2), C (Figura 3), F (Figura 4), N (Figura 5) y P (Figura 6). La revisión de estas figuras ayuda a comprender cómo funciona el proceso de distinguir entre las diferentes clases. Por ejemplo, el clasificador no toma muy en cuenta bigramas como 'feliz cumpleaños' en la categoría A pero les confiere la mayor importancia en la categoría de F, donde se englo-

Configuraciones			Conjunto de características					
Con Palabras Funcionales	Raíz	TF-IDF	1-grama	1-2-grama	1-3-grama	1-4-grama	1-5-grama	1-6-grama
✓	✓	✓	53.25 %	**64.18 %**	**65.50 %**	66.53 %	67.05 %	67.37 %
✓	✓	✗	53.73 %	61.20 %	64.76 %	66.05 %	66.73 %	67.00 %
✓	✗	✓	**53.81 %**	61.77 %	65.46 %	**66.64 %**	**67.28 %**	**67.52 %**
✗	✓	✓	52.32 %	59.86 %	63.79 %	64.89 %	65.45 %	65.73 %
✗	✓	✗	52.31 %	59.19 %	62.63 %	64.09 %	64.53 %	64.87 %
✗	✗	✓	53.29 %	61.00 %	64.33 %	65.34 %	65.96 %	66.20 %
✓	✗	✗	53.73 %	61.20 %	64.76 %	66.05 %	66.73 %	67.00 %
✗	✗	✗	53.22 %	60.49 %	62.86 %	64.45 %	65.15 %	65.40 %

Tabla 6: Promedios de exactitud de los clasificadores por conjuntos de características (n-gramas de caracteres) y diferentes configuraciones

Classe	Precisión	Exhaustividad	F_1
A	0.75	0.89	0.81
C	0.69	0.80	0.74
F	0.85	0.58	0.69
N	0.66	0.22	0.33
P	0.82	0.46	0.59
Promedio	0.75	0.59	0.63
Exactitud			0.74

Tabla 7: Reporte de clasificación para el modelo Support Vector Machine.

ban los tuits de felicitación. En la categoría C, en cambio, palabras como 'aprender', 'consejo' o 'éxito' se encuentran entre las más relevantes, mientras que los n-gramas preferidos para N no tienen en realidad relación con lo que normalmente entendemos por felicidad. Por último, en la categoría de la publicidad (Figura 6), son importantes hashtags como 'siguemeytesigo' o términos como 'coaching'.

Junto con eso, es fácil observar en cada figura cómo algunas de las características consideradas parecen no tener mucho sentido. Además, se puede inferir que un pre-procesamiento más preciso podría tener efectos positivos en el resultado final de clasificación. Finalmente, todos los experimentos y resultados están disponibles en el repositorio de GitHub[2].

Para finalizar esta sección haremos una comparación entre el modelo que obtuvo mejores porcentajes en el rendimiento promedio y un modelo pre-entrenado. Para ello se ha considerado el modelo *BERT multilingual uncased* (Devlin et al., 2018) entrenado por cuatro épocas, y se ha aplicado al corpus con el siguiente preprocesamiento: eliminación de los hiperenlaces y eliminación de los signos de puntuación. Esto debido a que los modelos del lenguaje no necesitan extracción de raíces ni TF-IDF.

El modelo BERT obtuvo una exactitud final de 77.33 %. La exactitud por cada época de entrenamiento en el BERT se puede observar en la Tabla 8.

Época	Exactitud
1	0.75
2	0.79
3	0.77
4	0.77

Tabla 8: Reporte de exactitud obtenida por BERT para cada época de entrenamiento.

La Tabla 9 muestra el reporte de clasificación obtenido para el modelo BERT, mientras que la Tabla 7 muestra el reporte de clasificación para el modelo Support Vector Machine. La exactitud obtenida por BERT supera en 0.03 a la exactitud obtenida por el modelo SVM. Esto quiere decir que el modelo BERT tiene un mayor porcentaje de predicciones correctas. Sin embargo, si evaluamos la métrica de precisión podemos observar que el modelo SVM es 0.03 superior a BERT. La precisión por clase indica la proporción de instancias correctamente clasificadas en cada clase. Aquí podemos ver que aunque el modelo SVM no logra capturar muchas instancias de la clase **N** (solo el 22 % según la métrica de exhaustividad), las instancias clasificadas en esta clase son correctas en un 66 %, en comparación del 53 % del modelo de BERT. En contraparte, la métrica de exhaustividad nos dice que el modelo BERT logra identificar más instancias de cada clase en un promedio de 70 % respecto al modelo SVM que solo identifica un 59 % de instancias de cada clase, en promedio.

[2] https://github.com/GIL-UNAM/TwitterHappiness

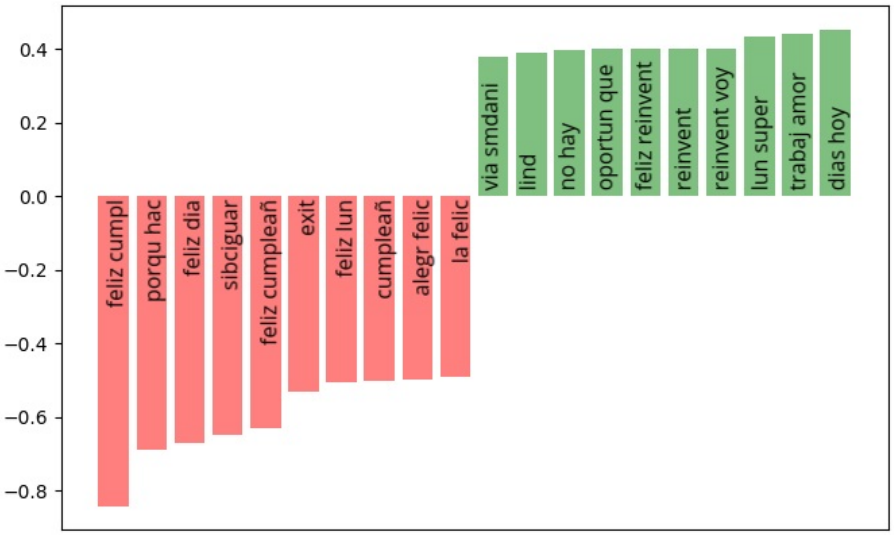

Figura 2: Top características de menor y mayor relevancia para la categoría A con SVM.

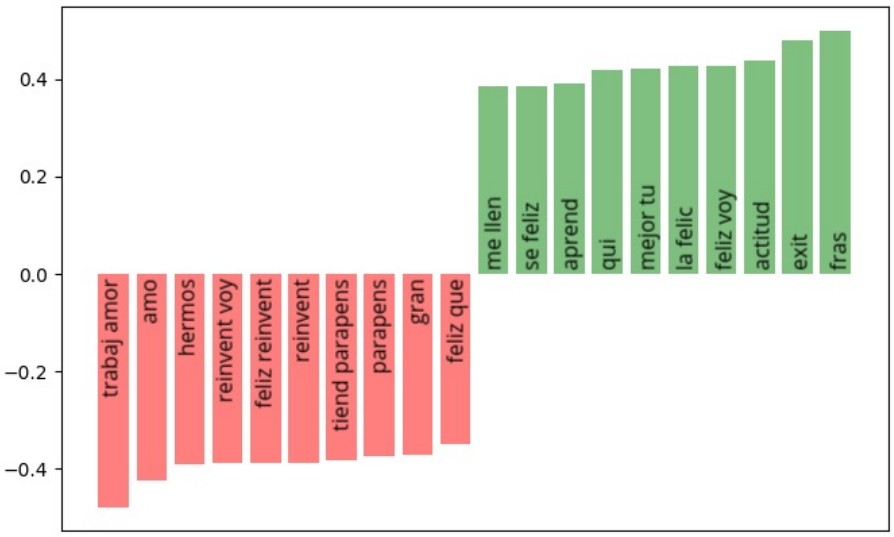

Figura 3: Top características de menor y mayor relevancia para la categoría C con SVM.

Classe	Precisión	Exhaustividad	F_1
A	0.83	0.84	0.83
C	0.77	0.85	0.81
F	0.73	0.74	0.74
N	0.53	0.43	0.48
P	0.74	0.65	0.69
Promedio	0.72	0.70	0.71
Exactitud			0.77

Tabla 9: Reporte de clasificación para el modelo BERT.

A pesar de los buenos resultados obtenidos en la evaluación del clasificador basado en el modelo *BERT multilingual uncased*, no es posible analizar directamente las características utilizadas por este modelo para realizar la clasificación (Yeh et al., 2020). Si bien existen técnicas para explicar cómo un clasificador realiza sus predicciones, basadas en el cálculo de un modelo local alrededor de la predicción a explicar (Rogers et al., 2021), estas no son necesariamente precisas (Pruthi et al., 2019). Serrano & Smith (2019) demuestran que los pesos de las atenciones no necesariamente corresponden con la importancia que tienen éstas dentro de los modelos.

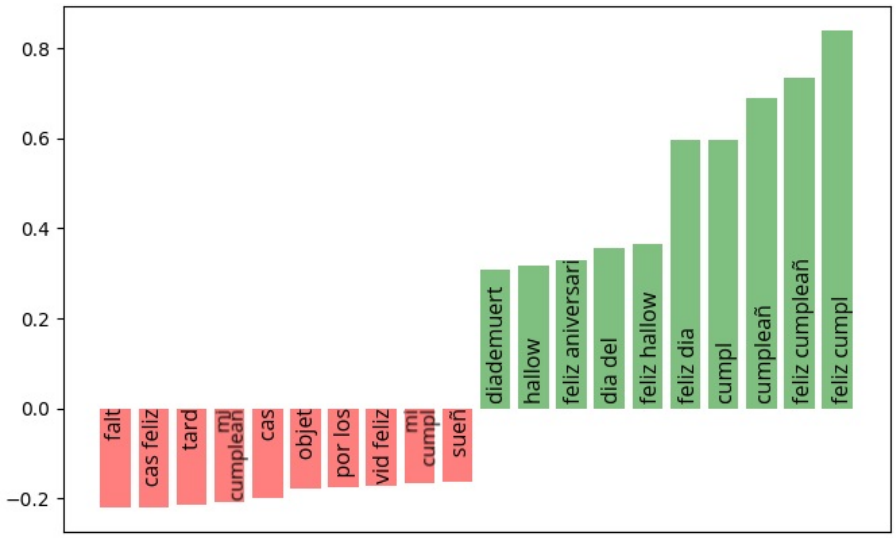

Figura 4: Top características de menor y mayor relevancia para la categoría F con SVM.

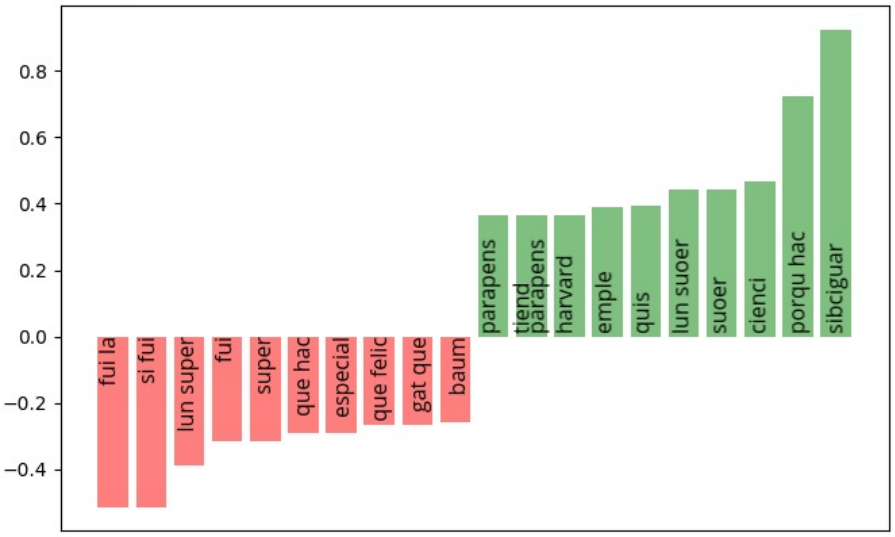

Figura 5: Top características de menor y mayor relevancia para la categoría N con SVM.

Por lo anterior, decidimos no realizar un análisis de las características del modelo *BERT multilingual uncased* sobre nuestro corpus. Sin embargo, este trabajo puede resultar interesante para entender mejor los modelos neuronales. Una posible ruta para este análisis es el modelo SELFEXPLAIN de Rajagopal et al. (2021), el cual incluye una capa interpretable globalmente que identifica los términos más relevantes dentro de un conjunto de entrenamiento, así como una capa interpretable local que permite calcular la contribución de cada input local respecto a una clase a predecir.

6. Conclusiones y trabajo futuro

Este artículo aporta una perspectiva adicional al estudio de las emociones en Twitter, en concreto centrándose en la felicidad. En general, los métodos para detectar la alegría basados en diccionarios han dado resultados muy consistentes. Nosotros nos preguntamos si siempre que se hace uso de este léxico el tuit denota realmente felicidad. Para la investigación nos hemos basado en un estudio anterior (Sidorov et al., 2016), donde se indican cinco categorías posibles que usan palabras relacionadas con el campo semántico que nos atañe: alegría, publicidad, felicitación, consejo y sarcasmo. Con este criterio se ha elabora-

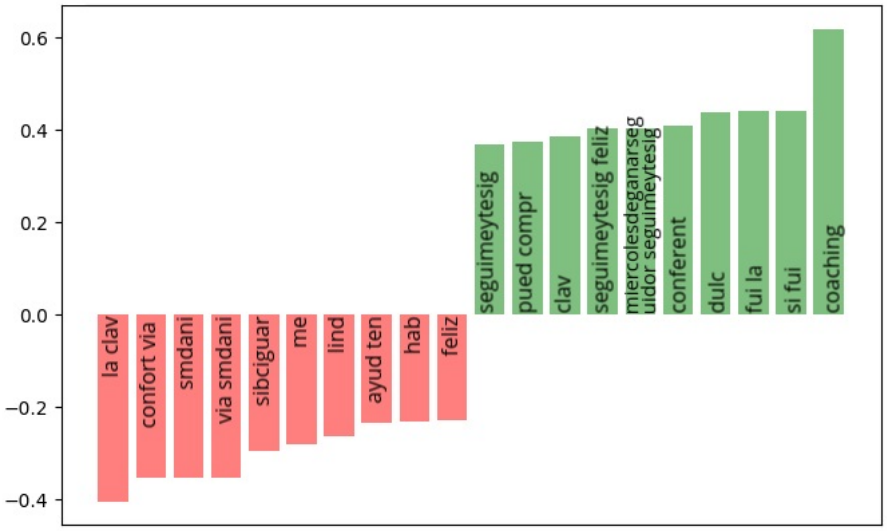

Figura 6: Top características de menor y mayor relevancia para la categoría P con SVM.

do un corpus obtenido con hashtags que remiten normalmente a la felicidad. Dicho corpus se ha etiquetado teniendo en cuenta estas categorías.

Los resultados explicitan que menos de la mitad de los tuits se refieren efectivamente a la emoción alegría, mientras un número elevado (21.5 %) se pueden considerar como consejos y reflexiones, e incluso casi el 14 % no suscita consenso entre los anotadores. El análisis léxico por categoría muestra que las variación en las palabras utilizadas en cada categoría es muy pequeña, de manera que el problema de distinguir entre unos significados o otros se convierte en una tarea complicada. Para abordarlo, se han implementado sistemas de aprendizaje automático con a) distintos tipos de pre-procesamiento y características; b) distintos clasificadores. Los mejores resultados se han obtenido con un pre-procesamiento que consiste en la inclusión de palabras funcionales, extracción de raíz y TF-IDF, con un sistema SVM. Pruebas adicionales han reportado que el uso de BERT produce una leve mejora, tanto en la exactitud como en F_1. Aún así, los mejores resultados de BERT no han sobrepasado el 78 % de exactitud.

Para el futuro, se propone el uso de redes neuronales para resolver este mismo problema, así como la extensión de la metodología al resto de emociones de Ekman.

El trabajo sobre identificación de emociones en redes sociales parece ser una tarea dura, ya que la denotación de los términos y el sentido general de los tuits no siempre se puede deducir directamente del significado connotativo del léxico utilizado. Este artículo es una primera aproximación a la clasificación de emociones según el sentido último que transmite el mensaje. Esta tarea se enmarca dentro del tratamiento computacional de procesos pragmáticos del lenguaje natural, entre los que se pueden contar el sentido figurado, la segunda intención o la necesidad de persuasión.

Reconocimientos

El presente trabajo se ha realizado con el apoyo de los proyectos CONACyT CB A1-S-27780, y DGAPA-UNAM PAPIIT números TA400121 y TA101722. Los autores agradecen al CONACYT por los recursos de cómputo brindados a través de la Plataforma de Aprendizaje Profundo para Tecnologías del Lenguaje del Laboratorio de Supercómputo del INAOE.

Referencias

Bakhtiyar, Syed, Indurthi Vijaysaradhi, Shah Kulin, Gupta Manish & Varma Vasudeva. 2019. Ingredients for happiness: Modeling constructs via semi-supervised content driven inductive transfer learning. Informe técnico. Centre for Search and Information Extraction Lab International Institute of Information Technology Hyderabad, India.

Banea, Carmen, Rada Mihalcea & Janyce Wiebe. 2011. Multilingual sentiment and subjectivity. En *Multilingual Natural Language Processing*, Prentice Hall.

Bansal, Mohit, Claire Cardie & Lillian Lee. 2008. The power of negative thinking: Exploiting label disagreement in the min-cut classification framework. En *International Conference on Computational Linguistics (COLING)*, 15–18.

Camacho Vázquez, Vanessa Alejandra, Grigori Sidorov & Sofía Natalia Galicia Haro. 2018. Automatic detection of negative emotions within a balanced corpus of informal short texts. *Cyberpsychology, Behavior, and Social Networking* 21(12). 781–787. 10.1089/cyber.2018.0207.

Chiorrini, Andrea, Claudia Diamantini, Alex Mircoli & Domenico Potena. 2021. Emotion and sentiment analysis of tweets using BERT. En *EDBT/ICDT Workshops*, vol. 2841, online.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee & Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR* abs/1810.04805.

Ekman, Paul. 1992. An argument for basic emotions. *Cognition and Eotion* 6. 169–200.

Ekman, Paul & Richard Davidson. 1994. *The nature of emotions: fundamental questions*. Oxford University Press.

Go, Alec, Richa Bhayani & Lei Huang. 2009. Twitter sentiment classification using distant supervision. Stanford University. https://cs.stanford.edu/people/alecmgo/papers/TwitterDistantSupervision09.pdf.

González, José Ángel, Lluís-F. Hurtado & Ferran Pla. 2021. TWilBert: Pre-trained deep bidirectional transformers for Spanish Twitter. *Neurocomputing* 426. 58–69. 10.1016/j.neucom.2020.09.078.

Gruzd, Anatoliy, Sophie Doiron & Philip Mai. 2011. Is happiness contagious online? a case of twitter and the 2010 winter olympics. En *44th Hawaii International Conference on System Sciences*, 1–9. IEEE. 10.1109/HICSS.2011.259.

Kim, Soo-Min & Eduard Hovy. 2006. Automatic identification of pro and con reasons in online reviews. En *COLING/ACL Main Conference and Poster Session*, 483–490.

Kim, Soo-Min & Eduard Hovy. 2007. Crystal: Analyzing predictive opinions on the web. En *Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 1056–1064.

Koppel, Moshe & Itai Shtrimberg. 2004. Good news or bad news? let the market decide. En *AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, 86–88.

Kumar, Akshi, Prakhar Dogra & Vikrant Dabas. 2015. Emotion analysis of twitter using opinion mining. En *8th International Conference on Contemporary Computing (IC3)*, 285–290. 10.1109/IC3.2015.7346694.

Mogilner, Cassie, Sepandar D Kamvar & Jennifer Aaker. 2011. The shifting meaning of happiness. *Social Psychological and Personality Science* 2(4). 395–402. 10.1177/1948550610393987.

Mohammad, Saif. 2012. #emotional tweets. En **SEM 2012: The First Joint Conference on Lexical and Computational Semantics*, 246–255.

Mohammad, Saif & Felipe Bravo-Marquez. 2017. Emotion intensities in tweets. *CoRR* abs/1708.03696. http://arxiv.org/abs/1708.03696.

Mohammad, Saif, Felipe Bravo-Marquez, Mohammad Salameh & Svetlana Kiritchenko. 2018. SemEval-2018 task 1: Affect in tweets. En *12th International Workshop on Semantic Evaluation*, 1–17. 10.18653/v1/S18-1001.

Mohammad, Saif & Svetlana Kiritchenko. 2018. Understanding emotions: A dataset of tweets to study interactions between affect categories. En *11th International Conference on Language Resources and Evaluation (LREC 2018)*, s.pp.

Morency, Louis-Philippe, Rada Mihalcea & Payal Doshi. 2011. Towards multimodal sentiment analysis: Harvesting opinions from the web. En *13th International Conference on Multimodal Computing (ICMI)*, 10.1145/2070481.2070509.

Naderi, Habibeh, Behrouz Haji Soleimani, Saif Mohammad, Svetlana Kiritchenko & Stan Matwin. 2018. DeepMiner at SemEval-2018 task 1: Emotion intensity recognition using deep representation learning. En *12th International Workshop on Semantic Evaluation*, 305–312. 10.18653/v1/S18-1045.

Plutchik, Robert. 1980. *A general psychoevolutionary theory of emotion* 3–33. Academic Press. 10.1016/B978-0-12-558701-3.50007-7.

Pruthi, Danish, Mansi Gupta, Bhuwan Dhingra, Graham Neubig & Zachary Lipton. 2019. Learning to deceive with attention-based explanations. *arXiv preprint arXiv:1909.07913* .

Rajagopal, Dheeraj, Vidhisha Balachandran, Eduard Hovy & Yulia Tsvetkov. 2021. Selfexplain: A self-explaining architecture for neural text classifiers. *arXiv preprint arXiv:2103.12279* .

Rogers, Anna, Olga Kovaleva & Anna Rumshisky. 2021. A Primer in BERTology: What We Know About How BERT Works. *Transactions of the Association for Computational Linguistics* 8. 842–866. 10.1162/tacl_a_00349.

Rosá, Aiala & Luis Chiruzzo. 2021. Emotion classification in Spanish: Exploring the hard classes. *Information* 12(11). 438. 10.3390/info12110438.

Serrano, Sofia & Noah A Smith. 2019. Is attention interpretable? *arXiv preprint arXiv:1906.03731* .

Sidorov, Grigori, Sofía Natalia Galicia Haro & Vanessa Alejandra Camacho Vázquez. 2016. Construcción de un corpus marcado con emociones para el análisis de sentimientos en twitter en español. *Revista Escritos BUAP* 1. 1–33.

Thomas, Matt, Bo Pang & Lillian Lee. 2006. Get out the vote: Determining support or opposition from congressional floor-debate transcripts. En *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 327–335.

Turney, Peter. 2002. Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews. En *40th Annual Meeting of the Association for Computational Linguistics*, 417–424. 10.3115/1073083.1073153.

Vilares, David, Miguel A Alonso & Carlos Gómez-Rodríguez. 2013. Clasificación de polaridad en textos con opiniones en español mediante análisis sintáctico de dependencias. *Procesamiento del Lenguaje Natural* 50. 13–20.

Wojatzki, Michael, Torsten Zesch, Saif Mohammad & Svetlana Kiritchenko. 2018. Agree or disagree: Predicting judgments on nuanced assertions. En *7th Joint Conference on Lexical and Computational Semantics*, 214–224. 10.18653/v1/S18-2026.

Yeh, Chih-Kuan, Been Kim, Sercan Arik, Chun-Liang Li, Tomas Pfister & Pradeep Ravikumar. 2020. On completeness-aware concept-based explanations in deep neural networks. *Advances in Neural Information Processing Systems* 33. 20554–20565.

Zeng, Zeyuan, Shaowu Zhang, Lu Ren, Hongfei Lin & Liang Yang. 2021. Senti-bsas: A bert-based classification model with sentiment calculating for happiness research. En *7th International Conference on Computing and Artificial Intelligence*, 272—-277. 10.1145/3467707.3467748.

Proposta recebida em Junho 2021 e aceite para publicação em Abril 2022.

Detecção de quebras em diálogos humano-computador

Human-computer dialogue breakdown detection

Leonardo de Andrade
Universidade de São Paulo
Escola de Artes Ciências e Humanidades

Ivandré Paraboni
Universidade de São Paulo
Escola de Artes Ciências e Humanidades

Resumo

Com o crescimento constante no uso de tecnologias de relacionamento com o consumidor na Internet, os sistemas de *chatbot* se tornaram onipresentes no processamento de linguagem natural (PLN) e áreas relacionadas. Apesar dos avanços significativos nos últimos anos, no entanto, sistemas desse tipo nem sempre fornecem resultados plausíveis e consistentes, em muitos casos levando a uma quebra no diálogo. Assim, há grande interesse em investigar as circunstâncias nas quais erros deste tipo são produzidos e, quando possível, aprimorar o projeto destes sistemas de modo a minimizar tais erros. Com base nestas observações, neste trabalho abordamos a questão da detecção automática de quebras em diálogos humano-computador apresentando três modelos que levam em consideração o histórico de diálogo para decidir quando ele possui maior probabilidade de culminar em uma quebra. Os modelos propostos exploram uma variedade de métodos de PLN recentes, e são avaliados tanto com base em um conjunto de dados de diálogos reais em português entre usuários humanos e sistemas de *chatbot* desenvolvido especificamente para este fim, como também utilizando *benchmarks* publicamente disponíveis para o idioma inglês.

Palavras chave

classificação textual, detecção de quebras em diálogos, chatbots

Abstract

With the steady growth in the use of consumer relationship technologies on the Internet, chatbot systems have become ubiquitous in Natural Language processing (NLP) and related fields. Despite significant advances in recent years, however, systems of this kind do not always deliver plausible, consistent results, in many cases leading to a dialogue breakdown. As a result, there is a growing interest in how to improve systems of this kind so as to minimise errors. Based on these observations, this work addresses the issue of automatic dialogue breakdown detection by presenting three models that take the dialogue history into account to decide when a conversation is likely to break. The models under consideration explore a range of recent NLP methods and are evaluated by using a purpose-built Portuguese dataset conveying real-world human-computer conversations, and also in publicly available benchmarks for the English language.

Keywords

text classification, dialogue breakdown detection, chatbots

1. Introdução

Chatbots são sistemas computacionais que interagem com usuários humanos usando língua natural, atuando como uma interface para provedores de dados e serviços (Brandtzaeg & Følstad, 2017). Sistemas deste tipo podem fazer uso de métodos de Geração de Língua Natural para produção de mensagens sem participação de autores humanos e altamente customizadas, ou métodos de Recuperação de Informação para selecionar mensagens a partir de uma base de textos predefinidos. É comum também que estes sistemas façam uso de avatares ou outros recursos com maior ou menor semelhança com um ser humano (Chiaráin & Chasaide, 2016), objetivando aumentar o nível de engajamento dos usuários.

Diálogos com *chatbots* podem ser imperfeitos, e em certos casos o usuário pode não receber a resposta que estava procurando e ser forçado a tentar outras formulações até que, eventualmente, obtenha a informação desejada ou desista da interação (Sandbank et al., 2018). Assim, define-se uma quebra em um diálogo como sendo um ponto específico em que o usuário humano não tem mais condições de prosseguir com o diálogo (Martinovski & Traum, 2003).

DOI: 10.21814/lm.14.1.354

Um exemplo de quebra deste tipo, adaptado a partir de exemplos do córpus a ser discutido na Seção 3, é apresentado a seguir.

Sistema
Olá, tudo bem? Como posso te ajudar?

Cliente
Olá, ontem paguei parcelado minha fatura em atraso, mas recebi sms que ainda está em atraso. Vc pode verificar?

Sistema
Não entendi sua pergunta! Por favor escreva com outras palavras

Cliente
Parcelamento da fatura

Sistema
Para parcelar sua fatura, faça...

Neste exemplo, observa-se que a partir de um erro de interpretação inicial o sistema acaba desviando a conversa para um tópico diferente do que era a intenção do cliente, o que pode culminar em uma quebra no diálogo mais adiante caso o cliente decida que não vale a pena continuar tentando. Antes que esse ponto seja alcançado, entretanto, o diálogo pode apresentar sintomas de desarranjo ou pequenos incidentes de insatisfação (neste caso, a própria falha de interpretação) que, de forma cumulativa, levam ao colapso final. Ao acompanhar esses incidentes menores e estudar as suas causas, podemos identificar oportunidades de melhoria na interação humano-computador e potencialmente aprimorar o sistema para que erros deste tipo não ocorram, ou ocorram com menor frequência.

Antecipar —e então tentar evitar— possíveis quebras em diálogos é uma tarefa importante para o aprimoramento de sistemas de *chatbot*, e uma linha de pesquisa ativa no Processamento de Língua Natural (PLN). O problema tem sido inclusive tema de desafios computacionais (ou 'shared tasks') da série *Dialogue Breakdown Detection Challenge* (DBDC) para os idiomas inglês e japonês (Higashinaka et al., 2017, 2019), e é destas competições que tomamos a própria definição do problema computacional a ser tratado, possivelmente pela primeira vez, em língua portuguesa. De acordo com esta definição, uma série de interações entre um *chatbot* e um usuário humano (como no exemplo acima) pode ser classificada de três formas: como uma situação que certamente levaria a uma quebra (B=*breakdown*), como uma situação que não levaria a uma quebra (NB=*no breakdown*), e ainda como um caso intermediário em que a sequência poderia ou não levar a uma quebra (PB=*possible breakdown*).

De modo geral, estudos existentes na área de detecção de quebra em diálogos levam em conta uma quantidade limitada de informação para decidir se há quebra ou não, muitas vezes considerando apenas um par pergunta-resposta de cada vez. Como alternativa a este tipo de abordagem, o presente trabalho apresenta três modelos de detecção de quebra em diálogos para o português que levam em conta o histórico (ou memória) da conversa, e que exploram diferentes métodos de sucesso recente em outras tarefas de PLN, a saber: o uso de *embeddings* estáticos (Mikolov et al., 2013; Pennington et al., 2014) e sensíveis ao contexto (Devlin et al., 2019), e o uso de modelos neurais baseados em *gate recurrent units* (GRUs). Para este fim, os modelos em questão fazem uso de um conjunto de diálogos reais produzidos por *chatbots* brasileiros, e quer foram coletados especificamente para este projeto. Além disso, como forma de demonstrar o poder de generalização dos modelos propostos, é conduzida também uma avaliação aos moldes das competições DBDC, utilizando para este fim *benchmarks* de diálogos em inglês publicamente disponíveis para este idioma.

O restante deste documento está organizado da seguinte forma. A Seção 2 discute trabalhos relacionados na área de detecção de quebras em diálogos. A Seção 3 descreve a construção do córpus de diálogos em português e os modelos de detecção de quebras propostos. A Seção 4 descreve o procedimento de avaliação destes modelos com base em diálogos em português e inglês, e a Seção 5 apresenta os resultados propriamente ditos. Finalmente, a Seção 7 resume as contribuições deste trabalho e oportunidades de melhorias futuras.

2. Trabalhos relacionados

No presente trabalho, a detecção de quebras em diálogos é vista uma tarefa de aprendizado de máquina supervisionado baseada em dados textuais (diálogos) anotados com pontos em que houve quebra nos moldes da série de desafios *Dialogue breakdown detection challenge*, ou DBDC (Higashinaka et al., 2016, 2017, 2019), conforme discutido na Seção anterior. Este problema se distingue, por exemplo, da detecção de quebras em diálogos em língua falada (Black & Eskenazi, 2009), da detecção de estados do diálogo (Williams et al., 2013), e também do caso de diálogos orientados a uma tarefa específica (Bear et al., 1992; Carpenter et al., 2001; Bulyko et al., 2005) por apresentar maior variedade de quebras possíveis (Higashinaka et al., 2016).

Uma parte considerável dos estudos de interesse para o presente estudo é assim organizada em torno da série de desafios DBDC e dos conjuntos de dados rotulados que estes eventos disponibilizam, denominados córpus DBDC3 e DBDC4[1]. Uma visão geral dos desafios DBDC e dos córpus a eles associados é apresentada na Seção 2.1. Modelos computacionais para detecção de quebras em diálogos humano-computador, utilizando estes ou outros córpus, são revisados na Seção 2.2.

2.1. Os desafios DBDC e córpus

Os desafios DBDC3 (Higashinaka et al., 2017) e DBDC4 (Higashinaka et al., 2019) são competições (ou *shared tasks*) de sistemas de detecção automática de quebras em diálogos entre humanos e *chatbots*. Nestes eventos foram produzidos dois córpus de mesmo nome que se tornaram influentes na área, e que consistem de coleções de diálogos em inglês e japonês rotuladas com informação de quebras (B), não-quebras (NB) e possíveis quebras (PB). Com o possível intuito de facilitar a condução da tarefa da competição, os rótulos das categorias B, BP e PB ocorrem de forma balanceada em ambos os córpus, e portanto estes conjuntos de dados não necessariamente correspondem a um uso normal de sistemas deste tipo. No presente trabalho, apenas as porções em inglês serão discutida, respeitando-se a mesma divisão de treino e teste proposta nas competições de origem.

O córpus DBDC3 (Higashinaka et al., 2017) foi rotulado por um grupo de 30 anotadores em múltiplas rodadas de análise de concordância entre juízes. A porção em inglês deste córpus é composta de quatro coleções de diálogos de propósito geral entre *chatbots* e voluntários humanos e/ou recrutados por *crowd sourcing*, a saber: TKTK-100, de 100 sessões do conjunto do WOCHAT TickTock (Yu et al., 2016); IRIS-100, de 100 sessões do conjunto do WOCHAT IRIS (Banchs & Li, 2012); CIC-115, de 115 diálogos do *Conversational Intelligence Challenge*[2]; e YI-100, de 100 diálogos com um robô do Instituto de Física e Tecnologia de Moscou[3]. A Tabela 1 apresenta estatísticas descritivas do córpus DBDC3 inglês em cada um destes conjuntos.

O desafio DBDC3 contou com oito equipes participantes, sendo que seis delas trabalharam exclusivamente com a porção de dados em inglês. Dentre estes, tiveram mais destaques os estudos de Iki & Saito (2017); Lopes (2017); Kato & Sakai (2017); Sugiyama (2019); Takayama et al. (2019), que são discutidos na Seção 2.2.

Para a edição seguinte do evento, denominada DBDC4 (Higashinaka et al., 2019), foi construído um novo córpus com características similares, ou seja, mantendo-se os mesmos dois idiomas e definições de classes, porém desta vez rotuladas por um time de apenas 15 anotadores. A porção em inglês do córpus DBDC4 consiste de diálogos produzidos pelo sistema IRIS (Banchs & Li, 2012), e diálogos produzidos por seis *chatbots* não especificados, denominados apenas como bot01..06, a partir do conjunto de dados ConvAI2[4]. A Tabela 2 apresenta estatísticas descritivas do subconjunto de desenvolvimento deste córpus, tal qual definido na competição.

O desafio DBDC4 contou com quatro equipes participantes, cujas abordagens e resultados são descritos por Sugiyama (2021); Shin et al. (2019); Hendriksen et al. (2021); Wang et al. (2019) e discutidos na Seção 2.2 a seguir. Todas equipes trabalharam com a porção de dados em inglês, e duas delas trabalharam também com o conjunto de dados em japonês (não considerado no presente trabalho).

2.2. Detecção automática de quebras em diálogos

A área de detecção automática de quebras em diálogos tem apresentado grande crescimento em anos recentes, possivelmente influenciado pela própria organização da série de desafios DBDC (Higashinaka et al., 2017, 2019). Alguns dos estudos deste tipo mais diretamente relacionados ao presente trabalho são sumarizados na Tabela 3, com informações sobre a língua-alvo dos diálogos considerados, o tipo de representação textual (e.g., *embeddings* de palavras, documentos ou sentenças, *part-of-speech* (POS) etc.), método de aprendizado de máquina (AM) adotado e, quando pertinente, a posição geral no ranque de sistemas participantes das competições DBDC3 e DBDC4 para as tarefas em inglês em primeira execução com base na medida F_1 relatada.

Os estudos aqui descritos foram identificados por meio de uma revisão exploratória da literatura, partindo-se dos próprios relatórios das competições DBDC, dos artigos publicados pelos seus participantes durante e após a participação no evento, bem como de suas próprias referências. Além disso, face ao grande número de submissões (cada participante podia submeter

[1] A edição mais recente do evento, denominada DBDC5, ainda não disponibilizou o conjunto de dados utilizado.

[2] https://convai.io/data/

[3] https://www.slideshare.net/sld7700/

[4] https://github.com/DeepPavlov/convai/tree/master/data

	TKTK	IRIRS	CIC	YI
Diálogos	210	210	225	210
Não quebra	38,6%	33,5%	29,0%	35,0%
Possível quebra	28,2%	28,4%	33,1%	37,7%
Quebra	33,2%	38,1%	37,9%	27,3%

Tabela 1: Estatísticas descritivas do córpus DBDC3 inglês em Higashinaka et al. (2017).

	bot1	bot2	bot3	bot4	bot5	bot6	IRIS
Total de diálogos	39	38	42	41	2	6	43
Não quebra	40,4%	40,8%	35,8%	39,9%	22,0%	16,4%	30,0%
Possível quebra	29,4%	26,8%	29,5%	29,4%	37,0%	22,6%	30,4%
Quebra	30,2%	32,4%	34,7%	30,7%	41,0%	61,0%	39,6%

Tabela 2: Total de diálogos e distribuição de frases do sistema por classe no córpus DBDC4 inglês (desenvolvimento) (Higashinaka et al., 2019).

até três execuções), foram selecionados apenas os sistemas de maior destaque em cada competição com base nos critérios de avaliação considerados.

Com base neste levantamento, observa-se uma predominância de estudos no idioma inglês, uso de *embeddings* de palavras, e métodos de aprendizado neural como LSTM e BERT. Os estudos de Iki & Saito (2017); Lopes (2017); Kato & Sakai (2017); Sugiyama (2019); Takayama et al. (2019) descrevem sistemas participantes da competição DBDC3 ou versões aprimoradas destes, e os estudos de Sugiyama (2021); Wang et al. (2019); Shin et al. (2019); Hendriksen et al. (2021) são relativos à competição DBDC4. Posteriormente, os estudos de Almansor et al. (2021); Ng et al. (2020b) apenas reutilizaram estes (e outros) conjuntos de dados de forma independente. Detalhes adicionais são discutidos a seguir.

2.2.1. Participantes da competição DBDC3

O estudo de (Iki & Saito, 2017) é motivado pela observação de que o histórico de diálogos humano-computador frequentemente inclui um grande número de palavras não observadas durante o treinamento do modelo, o que pode impactar a qualidade da conversa. Como forma de contornar esta dificuldade, é proposto o uso de redes neurais *End-to-End* do tipo MemN2N (Sukhbaatar et al., 2015) em conjunto com representações sentenciais baseadas em *embeddings* de caracteres, e uma rede do tipo CNN para o módulo de atenção. O sistema final, denominado Pleco, obteve os melhores resultados de medida F_1 na competição DBDC3 em inglês considerando-se a tarefa de detecção da classe quebra (B). Na tarefa de detecção de possível quebras ou quebras (PB+B), entretanto, o sistema ainda ficou (assim como todos os demais participantes) abaixo do *baseline* de classe majoritária da competição. Este sistema será utilizado como *baseline* também em nossos experimentos relacionados ao córpus DBDC3, descritos na Seção 3.

O trabalho de Lopes (2017) objetivou investigar a possível generalização do problema de detecção de quebras em diálogos orientados a tarefas ou não, comparando duas abordagens: uma baseada em um conjunto reduzido de atributos orientados a tarefas e classificadores SVM, e a outra (puramente textual) usando *embeddings* de sentença e RNNs. De modo geral, a abordagem não orientada à tarefa apresenta melhor desempenho, obtendo a segunda melhor medida F_1 para a tarefa em inglês da competição DBDC3, e a melhor medida de precisão dentre os sistemas participantes.

O estudo de Sugiyama (2017), originalmente submetido à competição DBDC3 apenas para a tarefa em japonês, é alargado em 2019 para contemplar também a tarefa em inglês. Neste caso, os modelos propostos objetivaram estimar o grau de adequação da transição de tópicos a cada par pergunta-resposta do diálogo, e obtiveram os melhores resultados da competição neste idioma. De forma mais específica, foram computadas diversas características textuais, como quantidade de palavras em comum entre pergunta e resposta, métricas de similaridade variadas, tamanho das sentenças em número de palavras e caracteres, quantidade de interações, *embeddings* de sentenças utilizando codificação do tipo seq2sec, contagem de termos interrogativos, distância em relação a perguntas similares, contagens IDF e palavras de conteúdo abstrato. Como métodos de aprendizado, foi em-

Referência	Língua	Representação textual	Método	DBDC3	DBDC4
Iki & Saito (2017)	En	*character emb.*	CNN	#1	
Lopes (2017)	En	*word/document emb.*	LSTM, SVM	#2	
Sugiyama (2017)	Jp	*sentence emb.*, POS, TF-IDF	*Ensemble*		
Kato & Sakai (2017)	En	*word emb.*, TF-IDF	Similaridade		
Takayama et al. (2017)	Jp	*word emb.*	LSTM, CNN		
Sugiyama (2021)	En	*word emb.*, POS, TF-IDF	BERT		#1
Wang et al. (2019)	En	*word/sentence emb.*	*RF*, LSTM		#2
Shin et al. (2019)	En	*sentence emb.*	BiLSTM		
Hendriksen et al. (2021)	En	*word emb.*	LSTM		
Almansor et al. (2021)	En	TF-IDF, sentimento	*Ensemble*		
Ng et al. (2020b)	En	*word emb.*	BERT		

Tabela 3: Estudos recentes de detecção de quebra em diálogos humano-computador.

pregado um *ensemble* do tipo pilha de regressores (*Stack regressors* (van der Maaten & Hinton, 2008)) baseado em *Random Forest* (RF), *Extra-trees* (ETR), *K-nearest Neighbor* (KNN), *Gradient Boosting* (GBR) e *Support Vector* (SVR). O regressor ETR é o utilizado no nível superior da pilha para combinar as predições dos demais modelos.

O estudo de Kato & Sakai (2017) segue a abordagem de Sugiyama (2017) e também utiliza regressores ETR e outros métodos para estimar a média e a variância da distribuição de quebras, e então derivar as probabilidades de quebra a partir dessas estimativas. Para cálculo da similaridade entre *embeddings* de duas sentenças, melhores resultados foram observados utilizando-se a similaridade de cosseno entre todos os pares de termos. Esta abordagem seria posteriormente aprimorada e reapresentada à competição DBDC4 com melhores resultados (Wang et al., 2019).

O estudo de Takayama et al. (2019) tem como foco a questão do viés de anotação de quebras em diálogos, e estende a submissão (Takayama et al., 2017) originalmente apresentada à competição DBDC3 para detecção de quebras em diálogos apenas em japonês. O estudo propõe uma abordagem para detecção de quebras que explora diferenças entre anotadores, na qual os dados de treinamento são agrupados de acordo com a distribuição de anotações, e então utilizados para treinar detectores específicos para cada agrupamento. A classificação é realizada com uso de um modelo de *embeddings* de palavras e redes do tipo LSTM e CNN combinadas em uma arquitetura do tipo *Ensemble* para a predição final de quebras.

2.2.2. Participantes da competição DBDC4

O estudo de Sugiyama (2021) apresenta uma abordagem que combina atributos tradicionais de diálogo propostos em estudos prévios (Sugiyama, 2017, 2019) e outras, e modelo de língua pré-treinado BERT (Devlin et al., 2019). A proposta apresentou o melhor resultado de medida F_1 global (considerando possíveis quebras e quebras, ou PB+B) e a melhor acurácia da competição DBDC4 para o inglês, e os melhores resultados globais de classificação para o japonês. Este sistema, denominado NTTCS19, será utilizado como *baseline* também em nossos experimentos relacionados ao córpus DBDC4, descritos na Seção 3.

O estudo por Wang et al. (2019) estende a abordagem de Kato & Sakai (2017), originalmente apresentada na competição DBDC3, para a edição DBDC4 com diversas melhorias, incluindo a substituição do regressor ETR por *Random Forest*, e a predição direta das probabilidades dos rótulos em vez de estimar sua média e variância. A proposta utiliza um modelo de LSTM adaptado de Lopes (2017) com uso de uma CNN adicional para extração de características. Dentre várias arquiteturas consideradas, uma solução baseada em um *ensemble* de árvore de decisão e múltiplos modelos do tipo LSTM apresentou o segundo melhor resultado de medida F_1 em primeira execução na competição DBDC4 em inglês.

O estudo de Shin et al. (2019) utiliza redes bidirecionais LSTM (BiLSTM) com mecanismo de atenção global e *embeddings* BERT para detecção de quebras no córpus DBDC4 inglês, além de um mecanismo de atenção local para lidar com casos de quebra raros. Os melhores resultados são observados na detecção de quebras próximas ao fim do diálogo, sugerindo que a disponibilidade de mais informação contextual facilita a tarefa. Apesar do uso de métodos mais sofisticados do que os de vários outros participantes, entretanto, o sistema não alcançou resultados competitivos.

O estudo de Hendriksen et al. (2021) compara uma gama de modelos LSTM (*vanilla*, empilhado e bidirecional) e tipos de *embeddings* (Word2Vec e GloVe de diferentes origens) para detecção de quebras no córpus DBDC4 inglês. Os melhores resultados foram observados na configuração que usa a LSTM do tipo *vanilla* com *embeddings* GloVe *Common Crawl*, mas ainda assim inferiores aos de outros sistemas participantes.

2.2.3. Outras abordagens

Posteriormente às competições DBDC3/4, dois estudos relacionados são ainda dignos de nota. O estudo de Almansor et al. (2021) utiliza métodos de análise de sentimentos para detectar mudanças indicativas de quebra ou possível quebra em diálogos em sistemas de atendimento ao consumidor. O modelo proposto utiliza um léxico afetivo e contagens TF-IDF para classificar o sentimento associado a cada interação como sendo positivo, neutro ou negativo, utilizando um *ensemble* de classificadores do tipo *Multinomial Naive Bayes*, *Bernoulli Naive Bayes*, regressão logística e SVM. Resultados observados no córpus DBDC3 são superiores aos do *baseline* CRF da competição para o caso de quebra individual, mas ainda inferior no caso de soma das quebras e possíveis quebras.

Finalmente, o estudo de Ng et al. (2020b) investiga o uso de métodos de aprendizagem semi-supervisionada para aprimorar a detecção de quebras em diálogos, incluindo pré-treinamento contínuo em um conjunto de dados da rede social Reddit e um método de aumento de dados baseado em múltiplas dobras (Ng et al., 2020a). O conjunto de dados aumentado é utilizado em um modelo de classificação composto de um módulo BERT e um classificador *Multilayer Perceptron* (MLP). O modelo proposto obteve os melhores resultados na recente competição DBDC5[5], sendo 12% superior aos sistemas de *baseline* e outros participantes. Os resultados para o córpus DBDC4 não são entretanto diretamente comparáveis com os de outros sistemas porque a métrica de avaliação utilizada foi a medida F_1 da classe majoritária, e não das classes quebra e quebra + possível quebra originalmente adotadas por Higashinaka et al. (2019).

3. Materiais e métodos

O presente trabalho consiste da criação de um novo conjunto de dados em português brasileiro contendo diálogos humano-computador rotulados com informação de quebras, e da proposta e avaliação de três novos modelos de detecção de quebras em diálogos humano-computador que levam em conta o histórico (ou memória) da conversa. Estes dois itens são descritos individualmente a seguir, e o código desenvolvido para este fim encontra-se disponível para reúso[6].

3.1. Construção do córpus DBDBR português

Conforme discutido na Seção 2.2, existem conjuntos de dados publicamente disponíveis para estudo de problemas de detecção de quebras em diálogos nos idiomas inglês e japonês. No caso do idioma português, entretanto, não foram identificados recursos semelhantes. Além disso, observa-se que a anotação de grandes massas de dados deste tipo representa um custo considerável, tipicamente envolvendo um grande número de juízes e problemas de concordância. Com base nestas observações, optou-se assim por efetuar a construção de um córpus de diálogos reais em português entre humanos e *chatbots*, aqui denominado DBDBR, contendo quebras sinalizadas pelos próprios usuários e contornando assim a necessidade de anotação manual por terceiros.

O córpus DBDBR foi construído a partir de dados cedidos por uma empresa brasileira que comercializa um sistema de *chatbot* de atendimento para seus clientes, e com a qual o primeiro autor desta pesquisa mantém vínculo profissional. Por meio deste vínculo, foi obtida permissão de uso de parte dos dados gerados pelo sistema.

Além do conjunto de mensagens propriamente dito, cada diálogo pode incluir informações de *feedback* do usuário, que tem a opção de sinalizar sua insatisfação com uma resposta usando conceitos como 'Não foi isso que eu perguntei' e 'Resposta incorreta'. No presente trabalho, respostas associadas a este tipo de *feedback* negativo são tomadas (ainda que de forma aproximada) como pontos de quebra no diálogo, o que contorna a necessidade de uma anotação manual de alto custo. Diferentemente dos córpus DBDC3/4 descritos na Seção 2.1, entretanto, é importante observar que o presente método só oferece a distinção binária entre quebra e não quebra, ou seja, não existe a classe intermediária (possível quebra).

O córpus contempla dados em três domínios de diálogo que apresentaram o maior volume de interações em um período de dois meses: uma provedora de TV por assinatura, um banco e uma corretora. Diálogos na área de TV por assina-

[5] http://workshop.colips.org/wochat/@iwsds2020/shared.html

[6] https://github.com/landrady/DialogBreakdown

tura incluem dúvidas sobre instalação de equipamentos, agendamento de serviço técnico, contratação de pacotes e outros; diálogos da área bancária incluem dúvidas sobre empréstimos, prazos de cartões, limites, emissão de boleto e outros; e diálogos na área de corretora incluem dúvidas sobre investimentos, cancelamentos de operações, juros, uso de cartão de crédito etc. Em cada domínio, foram extraídos aproximadamente 10.000 diálogos de forma aleatória.

A Tabela 4 apresenta estatísticas descritivas de cada domínio do córpus coletado.

3.2. Modelos propostos

Seguindo o trabalho de Higashinaka et al. (2017, 2019), no presente trabalho a detecção de quebras nos córpus DBDC em inglês será definida como um problema de classificação ternária (quebra, possível quebra e não-quebra). Para o caso do córpus DBDBR em português, entretanto, a tarefa será definida como um problema de classificação binária dado que o córpus não possui rótulos de 'possível quebra'.

Em ambos os casos, a detecção de quebras será investigada considerando-se três estratégias que levam em conta um histórico mais amplo da conversa. Estas estratégias, denominadas RegW2V, RegBERT e GruGloVe, foram escolhidas por nos permitir explorar métodos alternativos de aprendizado (em especial, do tipo regressão e baseados em *Gate Recurrent Units*), e representações textuais de *embeddings* estáticos e sensíveis ao contexto, conforme detalhado a seguir. Exceto quando indicado, a definição dos valores ótimos para os hiper-parâmetros de cada modelo foi realizada por meio de um procedimento de *grid search* a ser detalhado na Seção 4.

O modelo RegW2V objetiva representar uma estratégia de classificação textual padrão baseada em *embeddings* estáticos aplicada à detecção de quebras em diálogos. Para este fim, o modelo utiliza um método de aprendizado do tipo *Gradient Boosting* (Friedman, 2001) e *embeddings* do tipo Word2Vec (Mikolov et al., 2013). O modelo recebe como entrada a concatenação de dois tipos de informação: (i) dois vetores representando o par de sentenças usuário-sistema, e (ii) dois vetores representando a memória do diálogo, que é o conjunto de perguntas ou respostas das cinco últimas interações usuário-sistema. As sentenças usuário-sistema (i) são representadas como vetores de contagens TF-IDF reduzidas com uso de *Principal Component Analysis* (PCA), e a memória (ii) é representada por um vetor de *embeddings* médios de 300 dimensões do tipo Skip-gram pré-treinados em português, obtidos de Hartmann et al. (2017), e inglês[7].

O modelo RegBERT objetiva constituir uma solução mais sofisticada para o problema na qual os *embeddings* estáticos são substituídos por *embeddings* sensíveis ao contexto. RegBERT é em grande parte semelhante a RegW2V, porém utilizando representações textuais do tipo BERT (Devlin et al., 2019) de 257 dimensões para o português, obtidos por Souza et al. (2020), e de 77 dimensões para o inglês[8]. Estes parâmetros, de grande impacto no tempo de treinamento de modelos BERT, foram escolhidos com base no tamanho médio das sentenças em cada córpus.

Finalmente, o modelo GruGloVe objetiva representar uma solução de tratamento mais tradicional para a noção de histórico da conversa, baseada na classificação de sequências implementada com uso de uma rede neural baseada em *Gate Recurrent Units* (GRUs) e *embeddings* estáticos do tipo GloVe (Pennington et al., 2014). A rede recebe como entrada dois tipos de informação: (i) sequências de *embeddings* de sentenças do usuário e do sistema, em ambos os casos compostos da média dos *embeddings* de 150 palavras, e (ii) características não textuais adicionais representando o tamanho médio das sentenças do usuário e do sistema e o identificador do diálogo. Estas informações são fornecidas à primeira camada da rede em blocos de 10 interações cada. A seguir, os blocos de sentenças do usuário e do sistema são combinados em dois vetores médios representando os dois participantes do diálogo (i.e., humano e computador), que são então fornecidos a três camadas recorrentes do tipo GRU concatenadas ao conjunto de características não textuais, e a duas camadas densas do tipo ReLU e Softmax, respectivamente. Esta arquitetura é ilustrada na Figura 1.

Dado que as classes são desbalanceadas, as probabilidades obtidas pelos modelos de regressão (no intervalo entre zero e um) são convertidas em classes nominais ordenadas (não quebra, possível quebra e quebra) com a definição de pontos de corte ajustados para cada intervalo. Mais especificamente, foi construído um modelo do tipo floresta aleatória para cada tarefa de classificação usando os parâmetros fixos de profundidade máxima 4, com até 3 folhas, pesos de classes balanceados e estado aleatório zero. A partir da árvore de cada tarefa, foram identificados os intervalos de probabilidade que correspondem a cada classe nominal.

[7] https://code.google.com/archive/p/word2vec/
[8] https://huggingface.co/bert-base-cased

Métricas	TV	Banco	Corretora
Quantidade de diálogos	9.990	9.988	9.973
Usuários únicos	9.936	9.813	9.080
Média de palavras do consumidor	4,64	3,74	9,7
Média de palavras do *chatbot*	21,12	18,60	18,49
Média de interações por diálogo	7,32	7,65	6,32
Tamanho do vocabulário do consumidor	11.088	9.728	15.902
Tamanho do vocabulário do *chatbot*	30.367	18.556	23.973
Quantidade de quebras	7.932	6.131	7.044
Quantidade de não quebras	140.126	146.832	121.968

Tabela 4: Estatísticas descritivas do córpus DBDBR português.

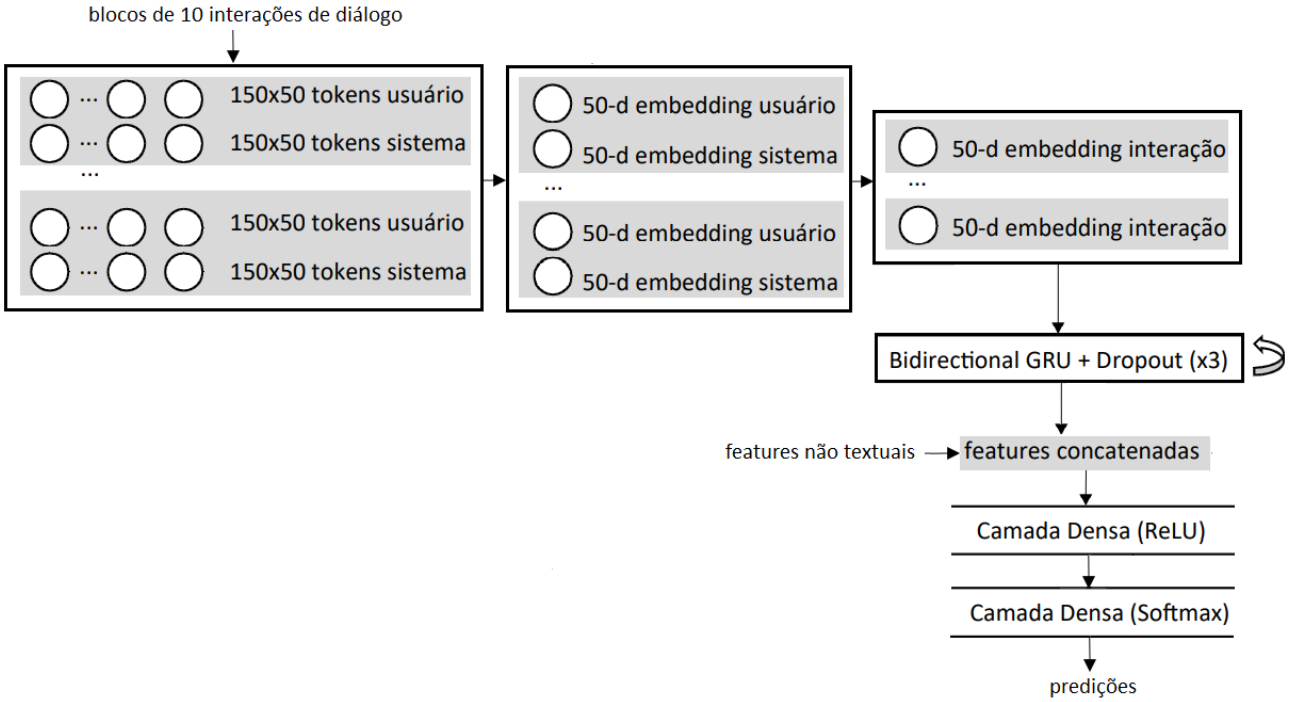

Figura 1: Arquitetura do modelo GruGloVe.

Um exemplo desta representação é apresentado na Figura 2, ilustrando o caso da árvore de decisão gerada para o modelo RegW2V com base nos dados do córpus DBDC3. Com base na probabilidade $X[0]$ do modelo de regressão subjacente, é exibido o número de instâncias das classes (*value*) não quebra, possível quebra e quebra, respectivamente, e o índice gini associado a cada nó da estrutura. O mesmo procedimento foi utilizado para a obtenção dos rótulos de classe nas demais tarefas de classificação aqui discutidas.

4. Avaliação

oram conduzidos experimentos para avaliar seu desempenho na tarefa de detecção de quebras nos diálogo no córpus DBDBR em português (cf. Seção 3.1) e nos córpus DBDC3 (Higashinaka et al., 2017) e DBDC4 (Higashinaka et al., 2019) do inglês. O objetivo do experimento foi assim o de identificar a melhor estratégia computacional para cada cenário de avaliação, e ilustrar a aplicação destes modelos a diálogos em português.

Para os córpus DBDC3/4, foi utilizada a mesma divisão de treino/teste seguida nas respectivas competições de modo a permitir uma comparação direta com sistemas existentes, e para o córpus DBDBR utilizou-se uma divisão aleatória à proporção 70/30. Em todos os casos, a porção de teste foi reservada para avaliação final dos modelos treinados.

Ao contrário dos conjuntos de dados em inglês, observa-se que o córpus português (DBDBR) é fortemente desbalanceado, com um número de não-quebras várias vezes superior ao número de quebras. Isso ocorre porque as avaliações destes diálogos são feitas pelos próprios usuários do

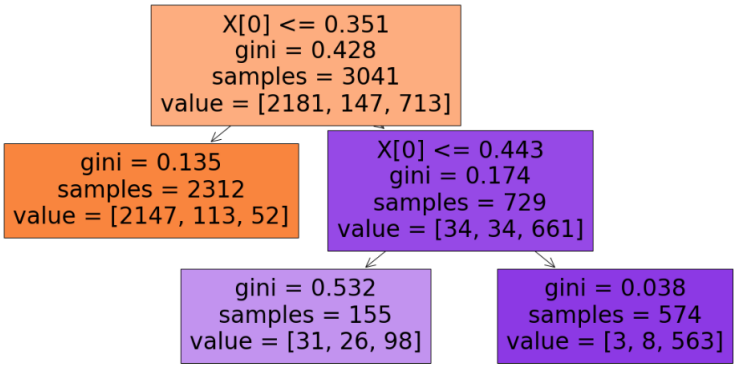

Figura 2: Árvore de decisão para o modelo RegW2V para o córpus DBDC3

serviço que, na maior parte das vezes, não fornece nenhuma resposta, o que configura o rótulo 'sem quebra'. Como forma de reduzir este desbalanceamento —permitindo assim a observação de diferenças mais expressivas entre os modelos avaliados— os experimentos a serem conduzidos utilizam apenas um subconjunto de cerca de 50% das instâncias da classe 'não quebra', selecionadas aleatoriamente a partir do córpus original. A Tabela 5 apresenta o número de instâncias para cada classe, conjunto e córpus.

Córpus	Conjunto	NB	PB	B
DBDBR	treino	134.108	-	14.774
	teste	57.476	-	6.333
DBDC3	treino	1.414	1.834	1.087
	teste	846	479	764
DBDC4	treino	1087	443	766
	teste	1239	446	481

Tabela 5: Instâncias de treino e teste para classes não quebra (NB), possível quebra (PB) e quebra (B).

No caso do modelo GruGloVe, a otimização foi feita considerando-se a variação de pesos validados pela função do erro médio quadrático (MSE). Para os modelos de regressão RegW2V e RegBERT, foi utilizado um otimizador de hiperparâmetros aleatórios com os atributos tamanho de janela (1 a 15), valor mínimo de DF para TF-IDF (DFmin, de 1 a 6), valor máximo de DF (DFmax, de 0,5 a 1), quantidade de estimadores (est, de 1 a 150), taxa de aprendizagem (lr, de 0,001 a 0,1) e profundidade máxima (P, de 3 a 6). Os valore destes hiper-parâmetros que obtiveram resultado ótimo de medida F_1 em cada conjunto de treinamento são sumarizados na Tabela 6.

Para fins de avaliação, os modelos propostos são comparados ao *baseline Conditional Random Fields* (CRF) de Higashinaka et al. (2017), e também com o sistema de melhor resultado em primeira execução em cada competição. No caso do conjunto DBDC3, isso corresponde ao sistema Pleco descrito por Iki & Saito (2017), e no caso do conjunto DBDC4 corresponde ao sistema NTTCS19 de Sugiyama (2021). Para o conjunto DBDBR, apenas o *baseline* CRF foi considerado. Os resultados dos modelos propostos foram computados utilizando-se a ferramenta de avaliação oficial de cada evento, e os resultados dos demais sistemas foram extraídos dos respectivos relatórios de Higashinaka et al. (2017, 2019).

Na avaliação propriamente dita foram consideradas as métricas de medida F_1 da classe 'quebra' $F_1(B)$ e, com exceção do córpus DBDBR (que não possui possíveis quebras), também a medida F_1 da soma das classes 'possível quebra' e 'quebra' $F_1(PB+B)$ propostas por Higashinaka et al. (2017). Todos os resultados são referentes à porção inédita de teste de cada córpus, à qual nenhum dos modelos teve acesso durante o treinamento.

5. Resultados

A seguir são apresentados os resultados individuais obtidos para cada córpus, seguidos da avaliação da sua significância estatística e de uma breve análise de erros.

5.1. Resultados principais

A Tabela 7 sumariza os resultados para a classe 'quebra' (B) no córpus DBDBR português (Seção 3.1), com o melhor resultado em destaque.

Observa-se que o modelo RegW2V é superior às alternativas consideradas, incluindo o próprio *baseline* CRF das competições DBDC3/4.

A Tabela 8 sumariza os resultados para as classes 'quebra' (B) e 'possível quebra' com 'quebra' (PB+B) no córpus DBDC3 inglês (Higashinaka et al., 2017). O melhor resultado de cada métrica é destacado.

Córpus	Modelo	janela	DFmin	DFmax	est	lr	P
DBDC3	RegW2V	5	4	0,92	15	0,001	4
	RegBERT	7	5	0,97	142	0,01	3
DBDC4	RegW2V	4	4	0,76	50	0,001	3
	RegBERT	4	4	0.76	50	0,001	3
DBDBR	RegW2V	6	2	0,91	100	0,1	5
	RegBERT	6	2	0,97	92	0,03	5

Tabela 6: Hiper-parâmetros ótimos para RegW2V e RegBERT em cada córpus.

Modelo	F_1(B)
Baseline CRF (Higashinaka et al., 2017)	0,53
RegW2V	**0,57**
RegBERT	0,56
GruGloVe	0,23

Tabela 7: Resultados de medida F_1 do córpus DBDBR português para a classe 'quebra'.

Modelo	$F_1(B)$	$F_1(PB+B)$
Baseline CRF (Higashinaka et al., 2017)	0,35	0,76
Pleco (Iki & Saito, 2017)	0,36	**0,87**
RegW2V	0,46	0,85
RegBERT	0,46	0,86
GruGloVe	**0,56**	0,84

Tabela 8: Resultados de medida F_1 do córpus DBDC3 inglês para a classe 'quebra' (B) e 'possível quebra' com 'quebra' (PB+B).

Nestes resultados observa-se que, para a métrica $F_1(B)$, o modelo GruGloVe supera as alternativas com ampla vantagem. No caso da métrica $F_1(PB+B)$, por outro lado, nenhum dos modelos propostos supera o sistema Pleco da competição DBDC3, ainda que a margem seja pequena (especialmente em relação ao modelo RegBERT).

Finalmente, a Tabela 9 sumariza os resultados para as classes 'quebra' (B) e 'possível quebra' com 'quebra' (PB+B) no córpus DBDC4 inglês (Higashinaka et al., 2019). O melhor resultado de cada métrica é destacado.

Modelo	F_1(B)	F_1(PB+B)
Baseline CRF (Higashinaka et al., 2017)	0,34	0,58
NTTCS19 (Sugiyama, 2021)	**0,46**	0,77
RegW2V	0,42	0,75
RegBERT	0,39	0,68
GruGloVe	0,41	**0,78**

Tabela 9: Resultados de medida F_1 do córpus DBDC4 inglês para a classe 'quebra' (B) e 'possível quebra' com 'quebra' (PB+B).

No caso da métrica $F_1(B)$, observa-se que os modelos propostos não atingem o resultado obtido pelo sistema NTTCS19 da competição DBDC4. Já no caso da métrica $F_1(PB+B)$, o modelo GruGloVe apresenta uma pequena vantagem sobre os demais.

Para análise de significância estatística, os três modelos propostos (RegW2V, RegBERT e GruGloVe) foram comparados ao *baseline* CRF[9] utilizado-se o método de *bootstrap* em Efron & Tibshirani (1994). De forma mais específica, para cada sistema um dos sistemas propostos (RegW2V, RegBERT e GruGloVe), e também para o *baseline* CRF, foram extraídas 100 amostras aleatórias de cada conjunto de predições com uma taxa de amostragem de 95%, e então foi calculada a medida F_1 média do sistema considerando a classe PB+B no caso da tarefa em inglês, ou apenas a classe B no caso da tarefa em português. Finalmente, os resultados de cada um dos sistemas propostos foram comparados aos resultados obtidos pelo baseline por meio de um teste-t. A Tabela 10 sumariza os testes realizados, na qual todas diferenças em relação ao *baseline* CRF são significativas para $p < 0,0001$.

Com base nestes resultados, contatou-se que os modelos propostos são significativamente superiores ao *baseline* CRF em todos os cenários, com exceção do modelo GruGloVe para o córpus DBDBR, em que foi observado um efeito significativo no sentido oposto.

5.2. Análise de erros

Como forma de identificar possíveis problemas de classificação e oportunidades de melhoria futura, foi realizada também uma breve análise de erros frequentes dos modelos RegW2V e GruGloVe sobre os córpus em inglês DBDC3 e DBDC4, já que estes apresentavam maior variedade do que a proporcionada pela rotulação binária do córpus

[9] A análise não inclui os sistemas Pleco (Iki & Saito, 2017) e NTTCS19 (Sugiyama, 2021) porque os resultados detalhados de suas predições não estão disponíveis, e porque não são aplicáveis ao córpus DBDBR.

	CRF	RegW2V		RegBERT		GruGloVe	
Córpus	F_1	F_1	teste t	F_1	teste t	F_1	teste t
DBDBR	0,725	0,743	138	0,734	76	0,402	2077
DBDC3	0,243	0,361	452	0,327	233	0,262	115
DBDC4	0,205	0,238	485	0,269	539	0,251	513

Tabela 10: Medida F_1 e estatísticas do teste t comparando o baseline CRF a cada um dos modelos propostos. Todas as diferenças em relação ao baseline são significativas para $p < 0,0001$.

em português. Para este fim, foram selecionados aleatoriamente 50 diálogos de cada córpus, totalizando 1034 interações humano-computador. Estas interações foram analisadas de forma empírica pelo primeiro autor deste estudo, que identificou quatro categorias de erros mais frequentes, aqui denominadas 'Erro de continuidade', 'Erro de anotação majoritária', 'Erro de saudação + pergunta' e 'Erro de quebras consecutivas'. A proporção de erros identificados em cada uma destas categorias é apresentada na Tabela 11, e detalhes adicionais são discutidos a seguir.

Erros de continuidade, exclusivos do modelo RegW2V, ocorrem quando o usuário continua o assunto de uma interação anterior porém o modelo identifica a não-quebra como sendo uma possível quebra, ou seja, 'esquecendo' o histórico do diálogo. Este tipo de problema foi melhor contornado com a classificação de sequências do modelo GruGloVe.

Erros de anotação majoritária representam os casos de maior ambiguidade na anotação de quebras presentes nos córpus DBDC3/4. Dado que os modelos consideram (assim como nas respectivas competições) o rótulo da classe como sendo aquele que tenha o maior número de anotações (ou votos) da equipe de juízes, observa-se que os modelos propostos tendem a classificar como possível quebra os casos em que a distribuição dos votos é mais balanceada, ou seja, quando não há uma tendência forte para quebra ou para não quebra.

Erros do tipo 'Saudação + Pergunta' são referentes ao uso combinado de uma saudação do usuário e de uma solicitação na mesma sentença, como em 'Olá, então quem você está visitando?'. Solicitações deste tipo são frequentemente respondidas pelo *chatbot* considerando-se apenas a saudação, e produzindo respostas como em 'Olá'. Todos estes casos constituem quebras de diálogo genuínas, mas tendem a ser classificadas apenas como possível quebra pelos modelos avaliados.

Finalmente, os erros do tipo 'Quebras consecutivas' são referentes ao efeito cumulativo de uma sequência de falhas no diálogo. Em casos deste tipo, os modelos propostos tendem a classificar incorretamente a sequência em sua totalidade mesmo quando parte das respostas era na verdade apropriada.

6. Discussão

Os experimentos realizados apresentam grande variação de resultados, o que era de certa forma esperado dada a variedade de conjuntos de dados, idiomas, definições de classe e métricas de avaliação. A seguir apresentamos de forma resumida algumas considerações a esse respeito.

Em primeiro lugar, observa-se que nas três tarefas abordadas, os melhores resultados foram obtidos por um dos sistemas propostos (RegW2V, RegBERT ou GruGloVe), ou por um sistema com resultados similares (i.e., sem diferença estatística significativa) em relação a estes. De forma mais específica, RegW2V obteve o melhor resultado para o córpus DBDBR, RegBERT ficou um ponto de medida F_1 abaixo do melhor modelo (Iki & Saito, 2017) para o córpus DBDC3, e GruGloVe obteve o melhor resultado para o córpus DBDC4.

Em segundo lugar, é interessante observar o papel de destaque dos modelos baseados em *transformers* do tipo BERT nestes experimentos. Não houve diferença significativa entre o melhor modelo de cada tarefa e RegBERT no caso dos córpus DBDBR e DBDC3, e somente na tarefa do córpus DBDC4 este modelo apresenta desvantagem real em relação às alternativas avaliadas. Ainda assim, cabe observar que o sistema NTTCS19 (Sugiyama, 2021), vencedor da competição DBDC4, é também baseado em um modelo de língua pré-treinado do tipo BERT.

Finalmente, embora as tarefas para o português e inglês não sejam verdadeiramente comparáveis (já que utilizam córpus diferentes e modelam tarefas de classificação diferentes), é interessante observar que, considerando-se os valores médios de medida F_1 obtidos, a tarefa em português parece ser mais complexa do que suas contrapartidas em inglês. Enquanto o melhor resultado de medida F_1 para o córpus DBDBR português foi 0,57, para as tarefas DBDC3 e

Categoria de erro	RegW2V # de erros	RegW2V % de erros	GruGloVe # de erros	GruGloVe % de erros
Erro de continuidade	23	3,8%	0	0,0%
Erro de anotação majoritária	161	26,9%	145	25,1%
Erro de saudação + pergunta	50	8,4%	61	10,5%
Erro de quebras consecutivas	99	16,6%	135	23,4%
Outros	265	44,3%	237	41,0%
Total de erros	598		578	

Tabela 11: Número (#) e percentual de erros cometidos pelos modelos RegW2V e GruGloVe na classificação de quebras nos córpus DBDC3/4, por categoria de erro.

DBDC4 em inglês obteve-se F_1 máximo de 0,87 e 0,78, respectivamente, o que é de certa forma inesperado considerando-se que a tarefa em português era binária, e portanto mais simples do ponto de vista computacional. Uma possível explicação para esta discrepância pode estar ligada ao tipo de fenômeno representado pelos rótulos de cada córpus. Como os rótulos dos córpus DBDC3/4 foram obtidos por consenso de grandes equipes de anotadores, é possível que estes rótulos representem uma classe mais restrita de problemas de quebra em diálogo, e que um alto grau de consistência na anotação facilite a tarefa de classificação automática. No caso do córpus DBDBR, por outro lado, o uso de rótulos derivados das indicações fornecidas por usuários, e talvez o próprio uso de dados de diálogos reais, contempla uma gama possivelmente muito maior de motivações para a quebra no diálogo, e com alto grau de subjetividade. Embora esta complexidade adicional em certo sentido torne o problema computacional mais realista, é possível também que isso explique o menor desempenho de todos os modelos empregados na tarefa em português.

7. Considerações finais

Este trabalho apresentou uma investigação de métodos de detecção automática de quebras em diálogos humano-computador em português e inglês levando em conta o histórico (ou memória) da conversa para decidir se a ocorrência de uma quebra é ou não provável. Para este fim, fora propostos modelos que fazem uso de regressão e GRU bidirecional, e utilizando *embeddings* de palavra estáticos e contextuais. Além disso, foi construído um novo córpus em português composto de diálogos reais produzidos por *chatbots* brasileiros que é, até onde temos conhecimento, um recurso inédito na área para este idioma.

Os resultados obtidos variam conforme a classe e o conjunto de dados considerado, não havendo uma solução ótima única para todos os cenários de avaliação. Ainda assim, os resultados dos modelos propostos são de modo geral próximos ou superiores aos dos sistemas de *baseline* considerados, incluindo os melhores sistemas participantes das competições DBDC3/4.

O estudo realizado deixa uma série de oportunidades de melhorias e trabalhos futuros. Em especial, destacamos que um trabalho mais extenso de otimização do modelo GRU pode levar a resultados superiores aos atuais, assim como combinações de *embeddings* contextuais BERT com outros métodos de classificação além da regressão logística da proposta atual. Outras possibilidades incluem, por exemplo, o estudo de quebras de referências pronominais[10] e o uso de conhecimento autoral como características de personalidade[11] do usuário em auxílio à tarefa de detecção de quebras em diálogos.

No que diz respeito ao córpus em português utilizado, observamos que o presente trabalho concentrou-se apenas nas quebras identificadas automaticamente por terem sido sinalizadas pelos usuários do sistema. É bastante provável, entretanto, que estes diálogos contenham muitas outras quebras não sinalizadas, e que seria igualmente importante conhecer e tratar computacionalmente. Um trabalho de anotação desta natureza, aos moldes do desenvolvido nas competições DBDC para o inglês e japonês, é também deixado como trabalho futuro, assim como a própria tarefa de criação de uma versão anonimizada dos dados, a ser disponibilizada para futuras pesquisas na área.

Agradecimentos

O segundo autor contou com apoio da Universidade de São Paulo.

[10] Paraboni (1997); Paraboni & de Lima (1998).

[11] Silva & Paraboni (2018a,b); dos Santos et al. (2017).

Referências

Almansor, Ebtesam Hussain, Farookh Khadeer Hussain & Omar Khadeer Hussain. 2021. Supervised ensemble sentiment-based framework to measure chatbot quality of services. *Computing* 103. 491–507. 10.1007/s00607-020-00863-0.

Banchs, Rafael E. & Haizhou Li. 2012. IRIS: a chat-oriented dialogue system based on the vector space model. Em *ACL 2012 System Demonstrations*, 37–42.

Bear, John, John Dowding, & Elizabeth Shriberg. 1992. Integrating multiple knowledge sources for detection and correction of repairs in human-computer dialog. Em *30th Annual Meeting of the Association for Computational Linguistics*, 56–63. 10.3115/981967.981975.

Black, Alan & Maxine Eskenazi. 2009. The spoken dialogue challenge. Em *10th Annual Meeting of the Special Interest Group in Discourse and Dialogue (SIGDIAL)*, 337–340.

Brandtzaeg, Petter Bae & Asbjørn Følstad. 2017. Why people use chatbots. Em *International Conference on Internet Science (INSCI)*, 377–392.

Bulyko, Ivan, Katrin Kirchhoff, Mari Ostendorf & J. Goldberg. 2005. Error-correction detection and response generation in a spoken dialogue system. *Speech Communication* 45(3). 271–288.

Carpenter, Paul, Chun Jin, Daniel Wilson, Rong Zhang, Dand Bohus & Alexander I. Rudnicky. 2001. Is this conversation on track? Em *EUROSPEECH 2001 Scandinavia; 7th European Conference on Speech Communication and Technology and 2nd INTERSPEECH Event*, 2121–2124.

Chiaráin, Neasa Ní & Ailbhe Ní Chasaide. 2016. Chatbot technology with synthetic voices in the acquisition of an endangered language: motivation, development and evaluation of a platform for irish. Em *10th International Conference on Language Resources and Evaluation (LREC)*, 3429–3435.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee & Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. Em *Conference of the North American Chapter of the Association for Computational Linguistics*, 4171–4186.

Efron, Bradley & Robert Tibshirani. 1994. *An introduction to the bootstrap*. CRC Press.

Friedman, Jerome. 2001. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics* 29(5). 1189–1232. 10.1214/aos/1013203451.

Hartmann, Nathan, Erick Fonseca, Christopher Shulby, Marcos Treviso, Jéssica Rodrigues & Sandra Aluísio. 2017. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. Em *11th Brazilian Symposium in Information and Human Language Technology (STIL)*, 122–131.

Hendriksen, Mariya, Artuur Leeuwenberg & Marie-Francine Moens. 2021. LSTM for dialogue breakdown detection: Exploration of different model types and word embeddings. Em *Increasing Naturalness and Flexibility in Spoken Dialogue Interaction: 10th International Workshop on Spoken Dialogue Systems*, 443–453. 10.1007/978-981-15-9323-9_41.

Higashinaka, Ryuichiro, Luis Fernando D'Haro, Bayan Abu Shawar, Rafael E. Banchs, Kotaro Funakoshi, Michimasa Inaba, Yuiko Tsunomori, Tetsuro Takahashi & Joao Sedoc. 2019. Overview of dialogue breakdown detection challenge 4. Em *Dialog System Technology Challenge*, em linha.

Higashinaka, Ryuichiro, Kotaro Funakoshi, Michimasa Inaba, Yuiko Tsunomori, Tetsuro Takahashi & Nobuhiro Kaji. 2017. Overview of dialogue breakdown detection challenge 3. Em *Dialog System Technology Challenge*, em linha.

Higashinaka, Ryuichiro, Kotaro Funakoshi, Yuka Kobayashi & Michimasa Inaba. 2016. The dialogue breakdown detection challenge: Task description, datasets, and evaluation metrics. Em *10th International Conference on Language Resources and Evaluation (LREC)*, 3146–3150.

Iki, Taichi & Atsushi Saito. 2017. End-to-end character-level dialogue breakdown detection with external memory models. Em *Dialog System Technology Challenges Workshop*, em linha.

Kato, Sosuke & Tetsuya Sakai. 2017. RSL17BD at DBDC3: Computing utterance similarities based on term frequency and word embedding vectors. Em *Dialog System Technology Challenges Workshop*, em linha.

Lopes, José. 2017. How generic can dialogue breakdown detection be? the KTH entry to DBDC3. Em *Dialog System Technology Challenges Workshop*, em linha.

van der Maaten, Laurens & Geoffrey Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9(86). 2579–2605.

Martinovski, Bilyana & David R. Traum. 2003. Breakdown in human-machine interaction: the error is the clue. Em *ISCA tutorial and research workshop on Error handling in dialogue systems*, 11–16.

Mikolov, Tomas, Scott Wen-tau & Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 746–751.

Ng, Nathan, Kyunghyun Cho & Marzyeh Ghassemi. 2020a. SSMBA: Self-supervised manifold based data augmentation for improving out-of-domain robustness. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1268–1283. 10.18653/v1/2020.emnlp-main.97.

Ng, Nathan, Marzyeh Ghassemi, Narendran Thangarajan, Jiacheng Pan & Qi Guo. 2020b. Improving dialogue breakdown detection with semi-supervised learning. Em *34th Conference on Neural Information Processing (NeurIPS)*, em linha.

Paraboni, Ivandré. 1997. *Uma arquitetura para a resolução de referências pronominais possessivas no processamento de textos em língua portuguesa*. Porto Alegre: Pontifícia Universidade Católica do Rio Grande do Sul. Tese de Mestrado.

Paraboni, Ivandré & Vera Lucia Strube de Lima. 1998. Possessive pronominal anaphor resolution in Portuguese written texts. Em *17th international conference on Computational linguistics-Volume 2*, 1010–1014.

Pennington, Jeffrey, Richard Socher & Christopher Manning. 2014. GloVe: Global vectors for word representation. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. 10.3115/v1/D14-1162.

Sandbank, Tommy, Michal Shmueli-Scheuer, Jonathan Herzig, David Konopnicki, John Richards & David Piorkowski. 2018. Detecting egregious conversations between customers and virtual agents. Em *Conference of the North American Chapter of the Association for Computational Linguistics*, 1802–1811. 10.18653/v1/N18-1163.

dos Santos, Vitor Garcia, Ivandré Paraboni & Bárbara Barbosa Claudino Silva. 2017. Big five personality recognition from multiple text genres. Em *Text, Speech and Dialogue (TSD)*, 29–37. 10.1007/978-3-319-64206-2_4.

Shin, JongHo, Alireza Dirafzoon & Aviral Anshu. 2019. Context-enriched attentive memory network with global and local encoding for dialogue breakdown detection. Em *Workshop on Chatbots and Conversational Agent Technologies*, em linha.

Silva, Bárbara Barbosa Claudino & Ivandré Paraboni. 2018a. Learning personality traits from Facebook text. *IEEE Latin America Transactions* 16(4). 1256–1262. 10.1109/TLA.2018.8362165.

Silva, Bárbara Barbosa Claudino & Ivandré Paraboni. 2018b. Personality recognition from Facebook text. Em *13th International Conference on the Computational Processing of Portuguese (PROPOR)*, 107–114. 10.1007/978-3-319-99722-3_11.

Souza, Fábio, Rodrigo Nogueira & Roberto Lotufo. 2020. BERTimbau: pretrained BERT models for Brazilian Portuguese. Em *9th Brazilian Conference on Intelligent Systems (BRACIS)*, 403–417. 10.1007/978-3-030-61377-8_28.

Sugiyama, Hiroaki. 2017. Dialogue breakdown detection based on estimating appropriateness of topic transition. Em *Dialog System Technology Challenges Workshop*, em linha.

Sugiyama, Hiroaki. 2019. Empirical feature analysis for dialogue breakdown detection. *Computer Speech & Language* 54. 140–150. 10.1016/j.csl.2018.09.007.

Sugiyama, Hiroaki. 2021. Dialogue breakdown detection using BERT with traditional dialogue features. Em *Increasing Naturalness and Flexibility in Spoken Dialogue Interaction*, Springer. 10.1007/978-981-15-9323-9_39.

Sukhbaatar, Sainbayar, Arthur Szlam, Jason Weston & Rob Fergus. 2015. End-to-end memory networks. Em *Advances in Neural Information Processing Systems (NIPS)*, em linha.

Takayama, Junya, Eriko Nomoto & Yuki Arase. 2017. Dialogue breakdown detection considering annotation biases. Em *Dialog System Technology Challenges Workshop*, em linha.

Takayama, Junya, Eriko Nomoto & Yuki Arase. 2019. Dialogue breakdown detection robust to variations in annotators and dialogue systems. *Computer Speech & Language* 54. 31–43. 10.1016/j.csl.2018.08.007.

Wang, Chih-Hao, Sosuke Kato & Tetsuya Sakai. 2019. RSL19BD at DBDC4: Ensemble of decision tree-based and LSTM-based models. Em

4th Dialogue Breakdown Detection Challenge, em linha.

Williams, Jason, Antoine Raux, Deepak Ramachandran & Alan W. Black. 2013. The dialog state tracking challenge. Em *14th Annual Meeting of the Special Interest Group on Discourse and Dialogue (SIGDIAL)*, 404–413.

Yu, Zhou, Ziyu Xu, Alan W. Black & Alexander Rudnicky. 2016. Strategy and policy learning for non-task-oriented conversational systems. Em *17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 404–412. 10.18653/v1/W16-3649.

Proposta recebida em Setembro 2021 e aceite para publicação em Abril 2022.

Análise Semântica com base em AMR para o Português

AMR-based Semantic Parsing for the Portuguese Language

Rafael Torres Anchiêta
Instituto Federal do Piauí

Thiago Alexandre Salgueiro Pardo
Núcleo Interinstitucional de Linguística Computacional (NILC)
Instituto de Ciências Matemáticas e de Computação
Universidade de São Paulo

Resumo

A Representação Abstrata de Significado (no inglês, *Abstract Meaning Representation* —AMR) é um formalismo semântico projetado para capturar o significado de uma sentença, representando-a como um grafo direcionado acíclico de única raiz com nós rotulados (conceitos) e arestas (relações) entre os nós. Essa representação tem recebido bastante atenção da comunidade de Processamento de Língua Natural, pois muitos autores têm proposto vários modelos de analise semântica para produzir grafos AMR a partir de uma sentença, visando melhorar o entendimento da língua natural. Entretanto, a maioria desses modelos focam no inglês devido a falta de grandes *corpora* anotados para outras línguas, deixando uma lacuna entre o inglês e outros idiomas. A fim de superar esse problema, neste artigo, é realizada uma análise detalhada de vários analisadores AMR, adaptando três modelos diferentes para o português e propondo melhorias. Além disso, estendeu-se um analisador baseado em regras desenvolvido previamente para o português. Esses modelos foram avaliados sobre um *corpus* anotado para o português. Por fim, realizou-se uma análise detalhada de erros com o objetivo de identificar os maiores desafios para análise no português e obter *insights* que possam ajudar pesquisas futuras nesta área.

Palavras chave

representação abstrata de significado, análise semântica, Português

Abstract

Abstract Meaning Representation (AMR) is a semantic formalism designed to capture the meaning of a sentence, representing it as a single rooted directed acyclic graph with labeled nodes (concepts) and edges (relations) among them. This representation has received growing attention from the Natural Language Processing community as many authors have proposed several models to produce an AMR graph from a sentence, aiming to improve natural language understanding. However, most of these models have focused on the English language due to the lack of large annotated corpora for other languages, producing a gap between English and other languages. To overcome this issue, in this paper, we carried out a fine-grained analysis of several parsers, adapted three different models to Portuguese, and proposed some improvements. Furthermore, we extended a previous rule-based AMR parser designed for Portuguese. We evaluated these models on a manually annotated corpus in Portuguese. Then, we performed a detailed error analysis to identify the major challenges in Portuguese AMR parsing that we hope will inform future research in this area.

Keywords

abstract meaning representation, semantic parsing, Portuguese

1. Introdução

A semântica computacional é a área encarregada de estudar representações semânticas viáveis computacionalmente para expressões na língua humana (Jurafsky & Martin, 2009). Nesta área, um analisador semântico, também conhecido como *parser* semântico, é responsável por verter o conteúdo de um texto em uma representação semântica computacional de maneira automática. Normalmente, isso ocorre abstraindo fenômenos sintáticos do texto e identificando, por exemplo, os sentidos das palavras, entidades nomeadas, papéis semânticos e outras características semânticas, visando eliminar interpretações ambíguas do texto (Goodman et al., 2016).

O desenvolvimento de analisadores semânticos é motivado pela hipótese de que a semântica pode ser usada para melhorar muitas tarefas de Processamento de Língua Natural (PLN), tais como: sumarização automática (Liu et al., 2015; Hardy & Vlachos, 2018), geração automática de

DOI: 10.21814/lm.14.1.358

texto (Pourdamghani et al., 2016; Song et al., 2017, 2018), vinculação de entidades (Pan et al., 2015; Burns et al., 2016), detecção de paráfrase (Issa et al., 2018; Anchiêta & Pardo, 2020b), sistemas de perguntas e respostas (Mitra & Baral, 2016) e tradução automática (Song et al., 2019), entre outras, produzindo sistemas melhores e mais informados.

Uma representação semântica é um dos ingredientes mais importantes de um analisador semântico. Por isso, vários pesquisadores têm empregado bastante esforço na criação de representações semânticas, como, por exemplo: a tradicional Lógica de Primeira Ordem, detalhada por Pereira & Shieber (2002) e Jurafsky & Martin (2009), Redes Semânticas (Lehmann, 1992), *Universal Networking Language* (UNL) (Uchida et al., 2006), *Universal Conceptual Cognitive Annotation* (UCCA) (Abend & Rappoport, 2013) e, mais recentemente, *Abstract Meaning Representation* (AMR) (Banarescu et al., 2013).

AMR, em particular, tem ganhado muita atenção da comunidade científica devido a sua estrutura relativamente simples, mostrando relações semânticas entre conceitos através de um grafo direcionado, como exibido na Figura 1.

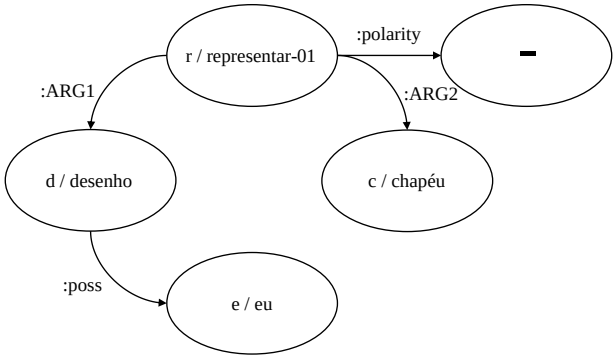

Figura 1: Exemplo de grafo AMR para a sentença "Meu desenho não representava um chapéu.", extraída do *corpus* composto pelo livro "O Pequeno Príncipe" (Anchiêta & Pardo, 2018a).

Na Figura 1, os nós são conceitos e as arestas são as relações semânticas. O conceito `representar-01` é a raiz do grafo e `:ARG1`, `:ARG2`, `:poss` e `:polarity` são relações do formalismo AMR.

A principal motivação para o uso de representações semânticas como AMR se deve ao fato de que elas apresentam a semântica de maneira explícita, permitindo realizar uma análise mais informada do conteúdo textual e dos resultados alcançados. Embora arquiteturas neurais, como as baseadas em *transformers*, tenham/estão atingindo resultados do estado da arte em diversas tarefas de PLN, o conhecimento semântico utilizado nessas modelagens está implícito em vetores densos (as conhecidas *word embeddings*), tornando a análise e interpretação dos resultados mais difícil. Há várias tarefas de PLN em que a semântica explícita pode ser necessária, como extração de informação, ferramentas de apoio à leitura e escrita e simplificação textual, entre outras. Também é interessante destacar as críticas que a semântica implícita tem recebido (veja, por exemplo, as críticas de Bender & Koller (2020)).

De acordo com Banarescu et al. (2013), a criação do formalismo AMR foi motivada pela necessidade de prover para a comunidade científica *corpora* com anotações semânticas relacionadas às tarefas tradicionais de PLN, tais como: reconhecimento de entidades nomeadas, anotação de papéis semânticos, desambiguação do sentido de palavras, resolução de correferência, e outras. A partir de *corpora* disponíveis, muitos analisadores AMR, de diferentes abordagens, foram desenvolvidos, por exemplo, baseados em grafo (Flanigan et al., 2014), em árvores de dependência (Wang et al., 2015b), em sistemas de transição (Damonte et al., 2017), em aprendizado profundo (van Noord & Bos, 2017; Lyu & Titov, 2018) e em *transformers* (Cai & Lam, 2020), entre outras.

A grande maioria desses analisadores está disponível apenas para o inglês devido à falta de grandes *corpora* anotados em outros idiomas. Essa escassez de recursos produz uma lacuna entre o inglês e outras línguas. Uma alternativa para preencher essa lacuna e criar aplicações mais efetivas de PLN é adaptar analisadores do inglês para outras línguas. Seguindo essa estratégia, Wang et al. (2018) adaptaram o analisador semântico de Wang et al. (2015b) para a língua chinesa, produzindo o primeiro analisador AMR para esse idioma.

Com base nesse contexto, este trabalho apresenta adaptações e melhorias de alguns analisadores AMR do inglês para o português. Além disso, investigou-se como as estratégias adaptadas se desempenham em uma língua com um *corpus* anotado pequeno. Realizou-se uma análise profunda de três analisadores AMR, identificando seus pontos fortes e suas fraquezas. Essa análise possibilitou a implementação de melhorias nesses analisadores. Junto com analisadores adaptados, propôs-se uma melhoria de um analisador desenvolvido previamente para a língua portuguesa.

Com objetivo de avaliar o desempenho dos analisadores, conduziu-se um experimento comparando grafos gerados automaticamente e grafos construídos manualmente, utilizando duas

métricas de avaliação de parsers AMR, Smatch (Cai & Knight, 2013) e SEMA (Anchiêta et al., 2019). Por fim, realizou-se uma análise detalhada de erros visando identificar os principais desafios para análise AMR na língua portuguesa, esperando auxiliar e fomentar pesquisas futuras nesta área.

No geral, este artigo faz as seguintes contribuições: (i) adaptação e melhoria de modelos de *parsing* AMR do inglês para o português, (ii) melhoria de um analisador desenvolvido para o português, (iii) uma visão geral dos analisadores AMR e (iv) uma análise detalhada de erros dos analisadores AMR adaptados.

O restante do artigo está organizado da seguinte maneira: a Seção 2 introduz os principais conceitos sobre o formalismo AMR; na Seção 3, são apresentados os *corpora* disponíveis nesta área; a Seção 4 oferece uma visão geral dos principais métodos de análise AMR; na Seção 5, é detalhada a adaptação de analisadores AMR para o português, bem como a implementação de melhorias; a Seção 6 reporta os experimentos conduzidos e os resultados obtidos; por fim, na Seção 7, conclui-se o artigo, indicando-se futuros direcionamentos.

2. Fundamentos de AMR

Abstract Meaning Representation (AMR) é um formalismo semântico produzido com o objetivo de capturar o significado de uma sentença, abstraindo elementos da estrutura sintática, como informação morfossintática e ordem das palavras (Banarescu et al., 2013). Esse formalismo descarta palavras que considera que contribuem pouco para o significado essencial da sentença, como artigos. Além disso, ele foca na estrutura predicado-argumento de uma sentença, conforme definido pelo PropBank (Kingsbury & Palmer, 2002; Palmer et al., 2005).

AMR pode ser representado como um grafo direcionado acíclico de raiz única com nós (conceitos) e arestas (relações) rotuladas. Os nós representam os principais eventos e entidades mencionados em uma sentença, enquanto as arestas representam o relacionamento semântico entre os nós, conforme apresentado na Figura 1. Os conceitos AMR podem ser concretos, compreendendo palavras em sua forma lexicalizada ("mulher", "homem"), *framesets* do PropBank ("representar-01") ou abstratos (como palavras chave especiais), que não correspondem a nenhuma unidade lexical da sentença, tais como *email-address-entity*, *percentage-entity*, *distance-quantity*, entre outros.

Além da estrutura em grafo, AMR pode ser representado em outras notações: tradicionalmente, na lógica de primeira ordem, para comparar e avaliar duas estruturas AMR, ou na notação PENMAN (Matthiessen & Bateman, 1991), para facilitar a leitura e a anotação humana. Na Figura 2, é exibido um exemplo de AMR na notação PENMAN (lado esquerdo) e um exemplo na lógica de primeira ordem (lado direito), respectivamente, para o grafo da Figura 1.

```
(r / representar-01
    :polarity -
    :ARG1 (d / desenho
        :poss (e / eu))
    :ARG2 (c / chapéu))
```

```
instance(r, representar-01)^
instance(d, desenho)^
instance(e, eu)^
instance(c, chapéu)^
polarity(r, '-')^
ARG1(r, d)^
poss(d, e)^
ARG2(r, c)
```

Figura 2: No lado esquerdo, notação PENMAN para AMR e, no lado direito, lógica de primeira ordem.

Outra característica do formalismo AMR é a natureza do relacionamento (alinhamento) entre as palavras de uma sentença e os nós do grafo. Nesse formalismo, não existe alinhamento explícito entre os nós do grafo e as palavras de uma sentença, ou seja, o alinhamento não faz parte da estrutura do AMR, embora ele seja importante para a tarefa de análise semântica.

Para avaliar estruturas AMR produzidas automaticamente por um *parser* semântico, utiliza-se tradicionalmente a métrica Smatch (Cai & Knight, 2013). Essa métrica computa o grau de sobreposição de uma estrutura AMR automática e uma de referência (produzida por humanos, normalmente), produzindo valores de precisão, cobertura e medida-f. Mais recentemente, Anchiêta et al. (2019) desenvolveram uma nova métrica para avaliar estruturas AMR, chamada SEMA. Essa métrica lida com alguns problemas da métrica Smatch (como não considerar a dependência entre elementos e distorcer o cômputo de alguns valores para certas situações de estruturação AMR), sendo mais rápida e robusta do que a Smatch. Além de serem usadas para a tarefa de avaliar analisadores AMR, essas métricas também são úteis para a tarefa de anotação semântica, permitindo, por exemplo, aferir a concordância entre humanos.

Na próxima seção, os principais *corpora* disponíveis na área são apresentados.

Língua	Disponível	Corpus	Treinamento	Desenvolvimento	Teste	Total
Inglês	LDC	LDC2013E117	8.684	1.085	1.085	10.854
		LDC2014T12	10.441	1.305	1.305	13.051
		LDC2015E86	16.833	1.368	1.371	19.572
		LDC2016E25 LDC2017T10	36.521	1.368	1.371	39.260
		LDC2020T02	55.635	1.722	1.898	59.255
Inglês	Público	O Pequeno Príncipe	1.274	145	143	1.562
		Bio AMR	5.452	500	500	6.452
Chinês	Público	O Pequeno Príncipe	1.274	145	143	1.562
Português			1.274	145	143	1.562
Espanhol				50		

Tabela 1: *Corpora* AMR

3. Corpora AMR

Existem vários *corpora* disponíveis para o inglês e algumas iniciativas para outras línguas.

Para o inglês, o *Linguistic Data Consortium* (LDC) é o principal responsável por disponibilizar *corpora* manualmente anotados. Os textos desses recursos são de diferentes domínios, como: notícias, fóruns de discussão, blogs e outros. Na Tabela 1, são detalhados os *corpora* disponíveis.

Os *corpora* LDC2015E86, LDC2016E25 e LDC2017T10 possuem as mesmas sentenças para os conjuntos de desenvolvimento e teste, e os *corpora* LDC2016E25 e LDC2017T10 são iguais[1]. A comunidade científica utiliza amplamente esses recursos, embora eles não estejam publicamente disponíveis. Até o momento, existem apenas dois recursos anotados públicos para o inglês[2]: o *corpus* produzido a partir do livro "O Pequeno Príncipe" e o Bio AMR. O primeiro contém o texto completo do livro escrito por Antoine de Saint-Exupéry, publicado em 1943 e traduzido para 300 línguas, enquanto que o segundo inclui textos de domínio biomédico, extraídos da PubMed[3].

Para *corpora* de outras línguas, existem algumas iniciativas, tais como em chinês (Li et al., 2016), português (Anchiêta & Pardo, 2018a; Sobrevilla Cabezudo & Pardo, 2019) e espanhol (Migueles-Abraira et al., 2018). Essas iniciativas anotaram uma versão de "O Pequeno Príncipe" em suas respectivas línguas. Para o espanhol, anotaram-se apenas 50 sentenças. Além dessas iniciativas monolíngues, existe um *corpus* multilíngue, o AMR 2.0–*Four Translations*[4]. Esse *corpus* possui 5.484 sentenças do *corpus* LDC2017T10 traduzidas para o italiano, espanhol, alemão e mandarim, ou seja, 1.371 sentenças para cada língua.

4. Trabalhos relacionados

Diversos estudos foram desenvolvidos para gerar estruturas AMR automaticamente a partir de sentenças. Aqui, são apresentados os principais estudos categorizados em seis classes: baseados em grafo, árvore, sistema de transição, gramática categorial combinatória, aprendizado profundo e *transformers*. Utilizando essas categorias, esta seção é dividida em três subseções: analisadores AMR para o inglês (4.1), analisadores para outras línguas (4.2) e, por último, um resumo dos analisadores (4.3).

4.1. Analisadores para o inglês

4.1.1. Métodos baseados em grafos

Métodos baseados em grafos identificam nós e computam pontuações de arestas para criar ligações entre os nós, adotando o algoritmo *maximum spanning connected subgraph*.

Flanigan et al. (2014) desenvolveram o primeiro analisador para o inglês, chamado JAMR. Os autores abordaram o problema em dois estágios: identificação de conceitos e identificação de relações. No primeiro estágio, eles atacaram o problema como uma tarefa de rotulação de sequência, adotando um modelo de semi-Markov para mapear blocos de palavras em uma sentença para nós no grafo. No estágio de identificar relações, os autores propuseram um algoritmo com o objetivo de encontrar um subgrafo

[1] O *corpus* LDC2017T10 está disponível para todos os inscritos no LDC, enquanto o *corpus* LDC2016E25 é limitado aos participantes do DEFT.

[2] https://amr.isi.edu/download.html

[3] https://www.ncbi.nlm.nih.gov/pubmed/

[4] https://catalog.ldc.upenn.edu/LDC2020T07

fortemente conectado sobre os conceitos identificados no primeiro estágio. Com o objetivo de desenvolver um algoritmo que aprenda esses dois estágios, os autores criaram um alinhador que mapeia as palavras de uma sentença aos nós do grafo correspondente. Com essa abordagem, o método desenvolvido alcançou 58% na métrica Smatch sobre parte do *corpus* LDC2013E117.

Werling et al. (2015) usaram o trabalho de Flanigan et al. (2014) como base e propuseram um novo método para identificação de conceitos, pois 38% das palavras do conjunto de desenvolvimento do *corpus* LDC2013E117 não são vistas durante o treinamento, tornando as abordagens baseadas em memorização frágeis. Os autores avaliaram a abordagem desenvolvida sobre o *corpus* LDC2014T12 e sobre parte do *corpus* LDC2013E117, atingindo 62% e 63,3% na métrica Smatch, respectivamente.

4.1.2. Métodos baseados em árvores

Abordagens baseadas em árvores iniciam a partir de uma árvore de dependência que é incrementalmente modificada para se tornar uma estrutura AMR.

Wang et al. (2015b) criaram um analisador AMR, chamado CAMR, que envolve dois passos. No primeiro passo, o modelo converte uma sentença em uma árvore de dependência, enquanto que o segundo passo transforma a árvore de dependência em um grafo AMR, realizando uma série de ações. Por exemplo, uma dessas ações é transformar a preposição "em" na relação AMR `:location`. Uma das principais vantagens desta abordagem é o uso de uma analisador de dependência que pode ser treinado em um grande *corpus*. O analisador CAMR obteve 63% na métrica Smatch sobre parte do *corpus* LDC2013E117. Em um trabalho posterior, Wang et al. (2015a) adicionaram uma nova ação para inferir conceitos abstratos. Além disso, eles incorporaram características mais ricas produzidas por analisadores auxiliares, tais como: anotador de papéis semânticos e resolvedor de correferência. Os autores reportaram uma melhoria de 7% na métrica Smatch.

Goodman et al. (2016) melhoraram o analisador proposto por Wang et al. (2015b), aplicando algoritmos de aprendizagem por imitação (Osa et al., 2018), visando reduzir ruídos produzidos pelo analisador CAMR. Com essa estratégia, os autores atingiram uma performance similar ao trabalho de Wang et al. (2015a) em parte do *corpus* LDC2013E117.

4.1.3. Métodos baseados em sistemas de transição

Um sistema de transição é uma máquina abstrata caracterizada por um conjunto de configurações (pilha de palavras parcialmente processadas, um *buffer* com palavras não vistas) e transições.

Zhou et al. (2016) propuseram um sistema de transição com o objetivo de aliviar a propagação de erros nos métodos baseados em grafo, executando conjuntamente tarefas de identificação de conceitos e relações em um modelo incremental. O modelo dos autores alcançou 67% na métrica Smatch sobre o *corpus* LDC2014T12.

Damonte et al. (2017) introduziram um analisador inspirado pelo sistema de transição `ArcEager` de Nivre (2004). A principal diferença entre eles é que o primeiro considera o mapeamento entre palavras de uma sentença e nós AMR, a não projetividade das estruturas AMR e nós de re-entrada (múltiplas arestas de entrada). Projetividade está relacionada à condição de não cruzamento de arestas, como apresentado na Figura 3. Com essa estratégia, o método atingiu 64% na métrica Smatch sobre o *corpus* LDC2014T12.

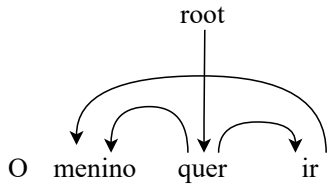

Figura 3: Um exemplo de não projetividade para a sentença "*O menino quer ir*". A aresta conectando a palavra "*ir*" para a palavra "*menino*" cruza outra aresta.

Ballesteros & Al-Onaizan (2017) utilizaram uma pilha de redes recorrentes para representar o estado do analisador. Além disso, eles adotaram algoritmos gulosos para tomar decisões sobre cada transição. Essa estratégia foi avaliada em parte do *corpus* LDC2013E117 e no *corpus* LDC2014T12, alcançando 69% e 64% na métrica Smatch, respectivamente.

Peng et al. (2018) apresentaram um sistema de transição que generaliza técnicas de análise de dependência com o objetivo de produzir grafos AMR. Para isso, eles usaram um *cache* de tamanho fixo, permitindo que o sistema crie arestas, ao mesmo tempo, para cada vértice no *cache*. Os autores avaliaram esse método no *corpus* LDC2015E86, atingindo 65% na métrica Smatch.

4.1.4. Métodos baseados em Gramática Categorial Combinatória

Gramática Categorial Combinatória (GCC) é um formalismo que provê uma interface transparente entre sintaxe e semântica (Steedman, 1996, 2001).

Artzi et al. (2015) mapearam sentenças para estruturas AMR em um processo de dois estágios. No primeiro, os autores adotaram um sistema de GCC para construir representações de cálculo lambda para aspectos composicionais do AMR. No segundo, eles propuseram um algoritmo de gramática de indução GCC para produzir grafos AMR a partir das representações da primeira etapa. Esse método atingiu 66,1% na métrica Smatch sobre parte do *corpus* LDC2013E117.

Misra & Artzi (2016) desenvolveram um analisador GCC que utiliza redes neurais, onde cada etapa do analisador é tratada como um problema de classificação multi-classe. Além disso, os autores propuseram um algoritmo iterativo que seleciona automaticamente a melhor análise na fase de treinamento. Eles avaliaram essa estratégia em parte do *corpus* LDC2014T12, alcançando 66,1% na métrica Smatch.

4.1.5. Métodos baseados em aprendizado profundo

Modelos de aprendizado profundo aprendem a produzir grafos AMR a partir de *corpora* anotados sem um processo manual de extração de características. A maioria dos trabalho adota uma estratégia chamada *sequence-to-sequence* (*seq2seq*) com redes *Bidirectional Long Short-Term Memory* (BiLSTM), convertendo um texto de entrada em um grafo AMR.

Peng et al. (2017) adotaram a estratégia *seq2seq* baseada no trabalho de Vinyals et al. (2015) e propuseram um linearizador e categorizador AMR, visando evitar dados esparsos. Essa estratégia obteve 52% na métrica Smatch sobre o *corpus* LDC2015E86.

van Noord & Bos (2017) utilizaram uma abordagem similar ao trabalho de Peng et al. (2017). Além disso, eles criaram um *corpus* adicional a partir dos analisadores JAMR e CAMR para o treinamento do modelo proposto. Essa abordagem obteve 71% na métrica Smatch sobre o *corpus* LDC2016E25.

Lyu & Titov (2018) introduziram um analisador neural que trata os alinhamentos entre as palavras de uma sentença e os nós do grafo como variáveis latentes e desenvolveram um modelo probabilístico que aprende de maneira conjunta conceitos, relações e alinhamentos. O analisador requer cinco diferentes redes LSTM para identificar os conceitos, relações, a raiz do grafo e os alinhamentos. Esse analisador atingiu 73,7% e 74,4% na métrica Smatch nos *corpora* LDC2015E86 e LDC2016E25, respectivamente.

4.1.6. Métodos baseados em transformers

Introduzido por Vaswani et al. (2017), *transformer* é uma arquitetura que evita recorrência, dependendo de mecanismos de atenção para obter relações entre a entrada e a saída da arquitetura.

Cai & Lam (2020) desenvolveram um modelo fim-a-fim que lida com a tarefa de análise AMR como uma série de decisões duplas sobre uma sentença de entrada, construindo o grafo AMR incrementalmente. Os autores também utilizaram *embeddings* contextuais do modelo de língua BERT (Devlin et al., 2019) para codificar as palavras da sentença de entrada, tendo um ganho de 2,9 pontos na métrica Smatch. A abordagem proposta atingiu 75,4% e 80,2% na métrica Smatch nos *corpora* LDC2014T12 e LDC2017T10, respectivamente.

Bevilacqua et al. (2021) trataram o problema de produzir um grafo AMR a partir de um texto como uma tarefa de transdução simétrica. Eles realizaram uma linearização cuidadosa nos grafos AMR e estenderam o modelo pré-treinado BART (Lewis et al., 2020) para produzir um grafo AMR. O método obteve 84,5% e 80,3% na métrica Smatch nos *corpora* LDC2017T10 e LDC2020T02, respectivamente.

4.2. Analisadores para outras línguas

Os trabalhos anteriores focaram na língua inglesa. Há relativamente poucos trabalhos para outras línguas, principalmente devido à falta de grandes *corpora* anotados, mas algumas iniciativas tentaram superar essa lacuna. Assim como foram organizados os trabalhos para a língua inglesa, aqui os trabalhos também estão organizados pelos seus métodos.

4.2.1. Métodos baseados em regras

Vanderwende et al. (2015) produziram um analisador que pode gerar grafos AMR para sentenças em francês, alemão, espanhol e japonês, onde anotações AMR não foram disponibilizadas. O método converte formas lógicas a partir de um analisador semântico (Vanderwende, 2015) em grafos AMR através de um conjunto de

Analisador	Ano	Abordagem	Corpus - Smatch (%)						
			2013N	2014N	2014	2015	2016	2017	2020
Flanigan et al. (2014)	2014	Grafo	58	-	-	-	-	-	-
Werling et al. (2015)	2015	Grafo	62.3	62.2	-	-	-	-	-
Wang et al. (2015b)	2015	Árvore	63	-	-	-	-	-	-
Wang et al. (2015a)	2015	Árvore	70	70	66	-	-	-	-
Artzi et al. (2015)	2015	GCC	-	66.3	-	-	-	-	-
Goodman et al. (2016)	2016	Árvore	70	-	-	-	-	-	-
Zhou et al. (2016)	**2016**	**Transição**	**71**	71	66	-	-	-	-
Misra & Artzi (2016)	2016	GCC	-	66.1	-	-	-	-	-
Peng et al. (2017)	2017	Aprendizado profundo	-	-	-	52	-	-	-
Damonte et al. (2017)	2017	Transição	-	-	-	64	-	-	-
Konstas et al. (2017)	2017	Aprendizado profundo	-	-	-	62.1	-	-	-
Foland & Martin (2017)	2017	Aprendizado profundo	-	-	-	70.7	-	-	-
Wang & Xue (2017)	2017	Árvore	-	-	68	68.1	-	-	-
Ballesteros & Al-Onaizan (2017)	2017	Transição	-	69	64	-	-	-	-
van Noord & Bos (2017)	2017	Aprendizado profundo	-	-	-	68.5	71	-	-
Peng et al. (2018)	2018	Transição	-	-	-	64	-	-	-
Vilares & Gómez-Rodríguez (2018)	2018	Transição	-	-	-	64	-	-	-
Lyu & Titov (2018)	**2018**	**Aprendizado profundo**	-	-	-	**73.7**	**74.4**	-	-
Guo & Lu (2018)	**2018**	**Transição**	-	**74**	68.3	68.7	-	69.8	-
Zhang et al. (2019)	2019	Transformer	-	-	70.2	-	-	76.3	-
Cai & Lam (2020)	**2020**	**Transformer**	-	-	**75.4**	-	-	80.2	-
Bevilacqua et al. (2021)	**2021**	**Transformer**	-	-	-	-	-	**84.5**	**83**

Tabela 2: Resultados dos analisadores AMR para o inglês.

regras. No entanto, os autores não avaliaram o método desenvolvido, pois não existiam *corpora* anotados para esse fim.

Anchiêta & Pardo (2018b) desenvolveram um analisador AMR baseado em regras para o português. Os autores propuseram regras genéricas para converter uma árvore de dependência com informações de papéis semânticos em um grafo AMR. Essa estratégia atingiu 53,3% na métrica Smatch no *corpus* anotado para o português (Anchiêta & Pardo, 2018a).

4.2.2. Métodos baseados árvore

Wang et al. (2018) adaptaram o analisador de Wang et al. (2015b) para o chinês e avaliaram o novo analisador em um *corpus* anotado para o chinês (Li et al., 2016), alcançando 58,7% na métrica Smatch.

4.2.3. Métodos baseados em sistemas de transição

Damonte & Cohen (2018) propuseram uma abordagem baseada em projeção de anotação, que envolve projetar a anotação de uma língua fonte em uma língua alvo. Usando o inglês como língua fonte, os autores produziram grafos AMR para o italiano, espanhol, alemão e chinês. No geral, essa estratégia obteve resultados distantes dos obtidos por analisadores desenvolvidos para o inglês, sendo 43% para o italiano, 42% espanhol, 39% alemão e 35% para o chinês, usando a métrica Smatch.

4.2.4. Métodos baseados em aprendizado profundo

Blloshmi et al. (2020) produziram um analisador AMR multilíngue, adotando técnicas de transferência de aprendizado. O analisador foi modelado como uma abordagem *seq2seq*, onde a camada de codificação é uma rede LSTM bidirecional e a camada de decodificação é uma rede LSTM unidirecional. Na camada de codificação, os autores incluíram *embeddings* multilíngue do BERT, a fim de produzir vetores contextualizados. O método gera grafos AMR para o italiano, espanhol, alemão e chinês, superando os resultados obtidos pelo trabalho de Damonte & Cohen (2018).

4.3. Sumário dos trabalhos

Na Tabela 2, é apresentado um resumo dos trabalhos mencionados para o inglês (uma vez que, para a mesma língua, a comparação é mais direta), organizados por ano de publicação. Os *corpora* 2013N e 2014N referem-se a seção de notícias de LDC2013E117 e LDC2014T12, respectivamente. Os *corpora* 2014, 2015, 2016, 2017 e 2020 são aqueles introduzidos na Seção 3. Na tabela, são destacados os melhores resultados obtidos pelos analisadores para cada *corpus*. É importante notar que os métodos baseados em sistema de transição seguidos pela abordagem baseada em árvore obtiveram melhores resultados para *corpora* pequenos, enquanto estratégias baseadas em aprendizado profundo e *transformers* tiveram melhor performance em *corpora* maiores.

Na próxima seção, serão detalhados os analisadores adaptados para o português, apresentando as melhorias propostas.

5. Adaptação de analisadores AMR para o português

O *corpus* de "O Pequeno Príncipe" anotado para o português (Anchiêta & Pardo, 2018a) tem 1.527 sentenças (alinhadas com a versão em inglês). Dessa forma, adaptaram-se abordagens baseadas em sistemas de transição e árvore, uma vez que essas estratégias atingiram bons resultados em *corpora* pequenos. Apesar disso, adaptou-se também um método baseado em aprendizado profundo, visando aferir seus resultados para o português. Por fim, implementaram-se melhorias no analisador de Anchiêta & Pardo (2018b) desenvolvido para o português.

Para as estratégias baseadas em sistemas de transição e árvore, adaptaram-se os analisadores de Damonte et al. (2017) (`AMREager`) e Wang et al. (2015b,a) (`CAMR`), respectivamente, pois eles são *open source* e necessitam apenas de pequenas modificações para reutilização para outra língua. Eles requerem um tokenizador, lematizador, etiquetador morfossintático (*tagger*), analisador de constituintes (*shallow parser*), analisador de dependência (*dependency parser*) e anotador de papéis semânticos. Para o português, essas ferramentas foram providas pelo Stanza (Qi et al., 2020), LX-Parser (Silva et al., 2010) e spaCy[5]. Além disso, os analisadores requerem alguns recursos léxicos, como: lista de países, estados e cidades, palavras negativas e *embeddings* pre-treinadas. Para o português, utilizaram-se as respectivas traduções das listas de palavras e *embeddings* pre-treinadas para o português (Hartmann et al., 2017).

Para treinar esses analisadores, é necessário alinhar os nós do grafo AMR com as palavras da respectiva sentença. Esses analisadores usam o alinhador `JAMR` (Flanigan et al., 2014) que produz alinhamentos como os mostrados na Figura 4.

```
:: alignments 0-1|0.0 1-2|0 2-3|0.1 3-4|0.3 5-6|0.2
(h / have-03
    :ARG0 (f / flower)
    :ARG1 (t / thorn)
    :purpose (s / spite)
    :mod (j / just))
```

Figura 4: Exemplo de alinhamento entre um grafo AMR e as palavras "*Flowers have thorns just for spite!*".

[5] https://spacy.io/

O formato do alinhamento é uma lista de *spans* separados por espaço com seu fragmento no grafo, onde cada nó é especificado por um descritor: `0` para o nó raiz, `0.0` para o primeiro filho do nó raiz, `0.1` para o segundo filho do nó raiz e assim por diante. No exemplo da figura, o *span* `0-1` (que é a palavra *Flowers*) está alinhado com o nó `0.0`. O `JAMR` foi desenvolvido para a língua inglesa, não possuindo bom desempenho para o português (Anchiêta & Pardo, 2020a). Por isso, uma primeira adaptação foi utilizar o alinhador desenvolvido por Anchiêta & Pardo (2020a) que foca na língua portuguesa. Apesar disso, esse alinhador não alinha relações de reentrada, ou seja, quando um nó participa de várias relações semânticas no grafo (assim como o `JAMR`). Para resolver essa questão, os nós que possuem relação de reentrada foram duplicados, como mostrado na Figura 5.

```
:: alignments 1-2|0 0-1|0.0
(s / say-01
    :ARG0 (h / he)
    :ARG1 h)
```

⇨

```
:: alignments 1-2|0 0-1|0.0 3-4|0.1
(s / say-01
    :ARG0 (h / he)
    :ARG1 (h1 / he))
```

Figura 5: Duplicação do nó `he` (lado direito) na relação de reentrada para a sentença "*He says to himself.*".

Além dessa melhoria no alinhamento, propôs-se um aprimoramento no modelo do analisador `CAMR`. Esse analisador adota um algoritmo baseado no perceptron para aprender características do conjunto de treinamento, usando a métrica Smatch para avaliar e escolher o melhor grafo AMR no conjunto de desenvolvimento. No entanto, como apontado por Song & Gildea (2019) e Anchiêta et al. (2019), a Smatch negligencia vários problemas de análise AMR, atribuindo pontuações altas para sentenças com diferentes significados. Por exemplo, as sentenças "*I have two houses*" e "*She bought three cars*" podem ser representadas como exibido na Figura 6. Usando a métrica Smatch para comparar esses dois grafos, ela retorna 0,29 de medida-f, ou seja, uma pontuação alta para sentenças com significados tão diferentes. Desse modo, o analisador `CAMR` pode escolher um grafo AMR pior devido à sobrevalorização da métrica Smatch. Apesar da métrica Smatch atribuir pontuações altas para sentenças com significados diferentes, os grafos AMR dessas sentenças compartilham informações em comum. Por exemplo, os grafos da Figura 6 compartilham as mesmas relações: `ARG0`, `:ARG1` e `:quant`, apesar dos conceitos (nós) entre as relações serem diferentes.

Com o objetivo de ajudar o analisador a escolher um grafo AMR melhor, substitui-se a métrica Smatch pela métrica SEMA (Anchiêta et al., 2019). SEMA é uma métrica mais rigorosa

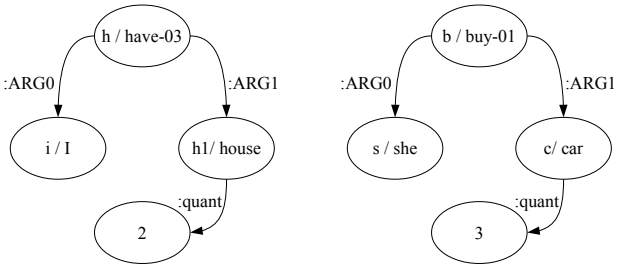

Figura 6: À esquerda, grafo AMR para a sentença "*I have two houses*"; à direita, grafo AMR para a sentença "*She bought three cars.*"

do que a Smatch, fazendo o analisador `CAMR` atualizar seus parâmetros e buscar um melhor grafo para uma sentença. Para o exemplo da Figura 6, SEMA retorna 0.00 de medida-f. A métrica SEMA compara dois grafos AMR em dois passos. No primeiro, SEMA tenta combinar apenas o nó raiz do grafo de hipótese com o nó raiz do grafo de referência. No segundo passo, a métrica combina o restante dos nós do grafo de hipótese com o restante dos nós do grafo de referência observando as arestas entre os nós, ou seja, se no grafo de hipótese e no grafo de referência houver arestas com o mesmo rótulo, o nó é computado como correto. No artigo de Anchiêta et al. (2019), mostrou-se que essa estratégia torna a SEMA mais rigorosa e mais rápida do que a Smatch.

As adaptações no alinhamento foram aplicadas nos analisadores `CAMR` e `AMREager`, enquanto que a SEMA foi aplicada apenas no `CAMR`.

Além dos analisadores acima, adaptou-se o analisador de van Noord & Bos (2017) (`NeuralAMR`), utilizando a ferramenta OpenNMT (Klein et al., 2017). Esse analisador adota uma abordagem *seq2seq* com uma rede bidirecional e um mecanismo de atenção geral (Luong et al., 2015). A Tabela 3 mostra os parâmetros do modelo.

Parâmetro	Valor	Parâmetro	Valor
Layers	2	RNN type	brnn
Nodes	500	Dropout	0.3
Epochs	20–25	Vocabulary	100–200
Optimizer	sgd	Max length	750
Learning rate	0.1	Beam size	5
Decay	0.7	Replace unk	true

Tabela 3: Parâmetros do modelo *seq2seq*.

Além desses parâmetros, o modelo aprende *embeddings* durante a fase de treinamento, ou seja, o modelo não usa *embeddings* pre-treinadas porque o *corpus* para o inglês é grande. Mais do que isso, os autores introduziram uma abordagem chamada "*super characters*", que é a combinação de palavras e caracteres na camada de entrada. Por exemplo, os autores transformaram relações AMR, como `:ARG0`, em atômicas ao invés de um conjunto de caracteres. Além disso, eles incorporaram classes morfossintáticas nas sentenças de entrada. A Figura 7 apresenta um exemplo dessa estratégia. Além disso, o grafo AMR é linearizado, sendo que as variáveis de cada conceito são removidas, pois o modelo não precisa aprender essas informações. As variáveis são recuperadas em uma etapa de pós-processamento. Por fim, esse modelo não precisa de um alinhador entre as palavras da sentença e um grafo AMR.

Para adaptar esse modelo, como o *corpus* de "O Pequeno Príncipe" é muito menor do que o usado pelos autores do `NeuralAMR`, adicionaram-se mais informações na estrutura "*super character*", tais como lema, relações de dependência e entidades nomeadas, com o objetivo de melhorar o analisador. Ademais, utilizaram-se *embeddings* pré-treinadas do português (Hartmann et al., 2017).

Mais do que a adaptação dos analisadores acima, melhorou-se um analisador AMR desenvolvido para o português (Anchiêta & Pardo, 2018b) (`RBAMR`). Esse analisador lida com o problema de gerar uma estrutura AMR aplicando um conjunto de regras genéricas sobre uma sentença de entrada pré-processada. Os autores desenvolveram seis regras para produzir as seguintes relações AMR: `named entity`, `:mod`, `:manner`, `:degree`, `:polarity` e `:time`. Embora essas relações sejam as mais frequentes no *corpus* de "O Pequeno Príncipe", os autores não trataram dois fenômenos essenciais que também aparecem com alta frequência: o verbo ser/estar e as conjunções. Dessa maneira, estendeu-se o analisador do português para lidar com esses fenômenos, adicionando-se as regras abaixo.

- **Verbo ser/estar.** Esta regra cria uma relação `:domain` em sentenças como ... ***ser/estar*** *<SUB>* e ... ***ser/estar*** *<ADJ>* quando o substantivo ou o adjetivo não for um *frameset* do PropBank. A Figura 8 mostra um exemplo dessa regra.
- **Conjunções.** Esta regra lida com dois tipos de conjunções: contrastiva e aditiva. Na primeira, produz-se o conceito `contrast-01` para as seguintes conjunções: *mas*, *enquanto*, *enquanto que*, *no entanto*, *entretanto* e assim por diante. Na segunda, criam-se os conceitos `and` ou `or`, como apresentado na Figura 9.

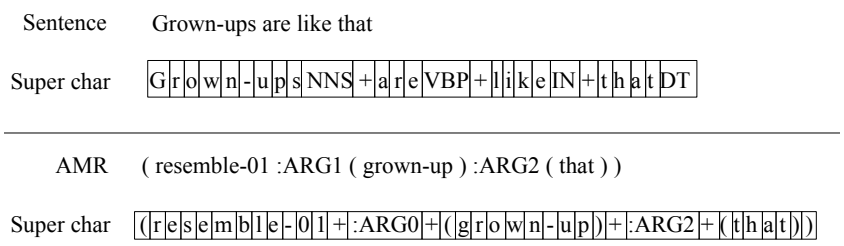

Figura 7: Sentença de entrada "*Grown-ups are like that*" com etiquetas e grafo AMR `(resemble-01 :ARG1 (grown-up) :ARG2 (that)))`. O símbolo + representa espaço.

Além dessas duas regras, desenvolveu-se um método de poda para os grafos AMR em uma etapa de pós-processamento com o objetivo de aumentar a qualidade dos grafos. O analisador `RBAMR` não mantém os traços de quais nós já foram produzidos, gerando redundância. Portanto, o método de poda remove todos os nós duplicados que possuem o mesmo nó pai.

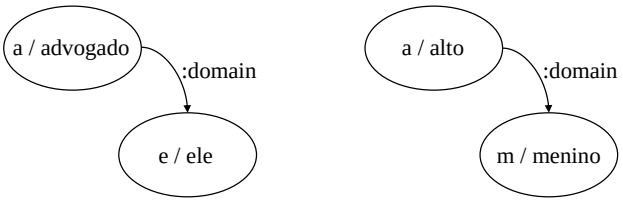

Figura 8: Regra para o verbo ser/estar para as sentenças: "*Ele é um advogado.*" (esquerda) e "*O menino é alto.*" (direita).

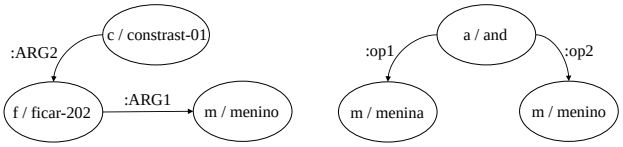

Figura 9: Regras para conjunções: conceitos `contrast-01` e `and`, respectivamente, para as sentenças: "*Mas o menino ficou.*" (esquerda) e "*A menina e o menino.*" (direita).

Como foi realizada uma duplicação nos nós com reentrada gerados pelos analisadores `CAMR` e `AMREager`, no pós-processamento recuperou-se a relação de entrada. A Figura 10 exemplifica esse processo.

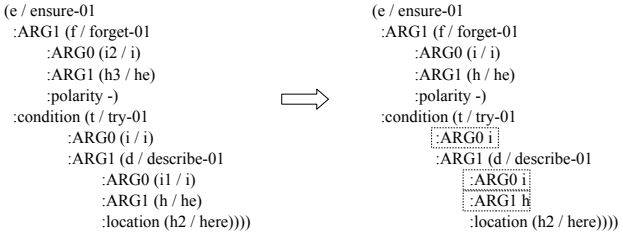

Figura 10: Um exemplo de nós com reentrada recuperados. Na esquerda, um exemplo com nós duplicados; na direita, os nós recuperados `i` e `he`.

Para o analisador `RBAMR`, aplicou-se um método de poda para remover nós redundantes que possuem o mesmo nó pai. Na Figura 11, é exibido um exemplo do método.

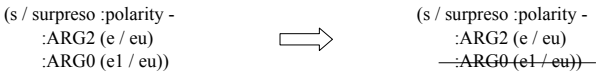

Figura 11: Na esquerda, nó redundante; na direita, nó podado `eu`.

No que segue, relata-se a avaliação dos analisadores, bem com uma detalhada análise de erros sobre um *corpus* anotado para o português.

6. Avaliação e resultados

Com o objetivo de avaliar os analisadores adaptados, conduziu-se um experimento sobre o *corpus* anotado de "O Pequeno Príncipe" (Anchiêta & Pardo, 2018a). O *corpus* está alinhado com a versão em inglês do livro, mantendo as divisões de treinamento, desenvolvimento e teste iguais ao do inglês[6], com 1.274, 145 e 143 sentenças, respectivamente. Além disso, separaram-se as sentenças do conjunto de teste pelo tamanho, pois, quanto maior a sentença, mais desafiadora é a análise, pois os erros produzidos em etapas de pré-processamento são propagados para a geração dos grafos. Portanto, calculou-se a média do tamanho das sentenças, obtendo um valor de 10,46 palavras por sentença, sendo 80 sentenças maiores que a média e 63 sentenças menores que a média. A separação por tamanho visou explicitar os desafios da análise.

Para avaliar os analisadores, as métricas Smatch e SEMA foram utilizadas. Nas Figuras 12 e 13, são apresentados os resultados das métricas para sentenças curtas e longas, respectivamente.

Nessas figuras, pode-se ver que o analisador `RBAMR` obteve os melhores resultados em ambas as métricas e tamanhos de sentença. O analisador atingiu 0,66 e 0,48 de medida-f para sen-

[6] https://amr.isi.edu/download.html

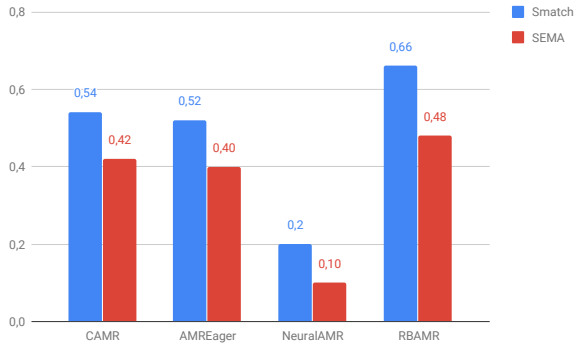

Figura 12: Resultados para sentenças curtas.

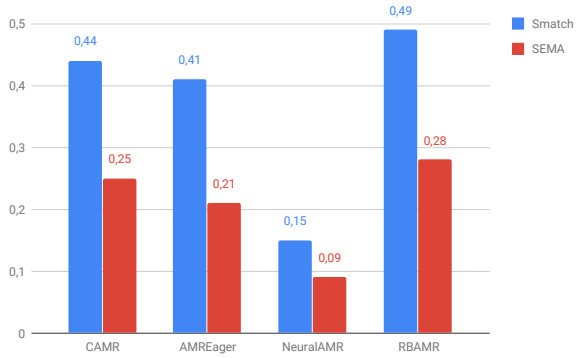

Figura 13: Resultados para sentenças longas.

tenças curtas na métrica Smatch e SEMA, respectivamente, e 0,49 e 0,28 para sentenças longas nas mesmas métricas. Os analisadores `CAMR` e `AMREager` tiveram um desempenho similar em ambas as métricas e tamanhos de sentenças, com o primeiro sendo levemente superior ao segundo. Por fim, o analisador `NeuralAMR` não alcançou bons resultados. Acredita-se que seja porque o *corpus* é pequeno.

A Tabela 4 apresenta uma análise detalhada dos resultados, mostrando as contribuições das adaptações propostas para cada analisador, exibindo valores de medida-f da Smatch e da SEMA. A partir dessa tabela, pode-se ver que as novas regras mais o método de poda melhoraram a versão anterior do `RBAMR` em 0,05 e 0,03 na métrica Smatch para sentenças curtas e longas, respectivamente. A melhoria no alinhamento também superou a versão original dos analisadores `CAMR` e `AMREager`. Além disso, adotar a métrica SEMA na fase de treinamento do `CAMR` produziu melhores resultados. É interessante perceber que modificações relativamente simples podem gerar resultados melhores.

Observando-se os grafos gerados pelos analisadores `CAMR` e `AMREager`, notou-se que um dos fatores para o resultado ser pior do que o `RBAMR` são os erros no alinhamento. Embora tenha sido utilizado um alinhador desenvolvido para o português, alguns fenômenos são difíceis de se lidar, como o sujeito oculto. Por exemplo, na sentença "*Preciso é de um carneiro.*", o sujeito '*eu*' não aparece na sentença, mas ele foi anotado, conforme exibido na Figura 14. Consequentemente, o nó `eu` não possui um correspondente na sentença.

```
:: alignments 0-1|0 4-5|0.1
Preciso é de um carneiro
(p / precisar-01
      :ARG0 (e / eu)
      :ARG1 (c / carneiro))
```

Figura 14: Alinhamento entre um grafo AMR e palavras para a sentença "*Preciso é de um carneiro.*".

Além da análise nos alinhamentos, observou-se também que o analisador `CAMR` produz relações `:null_edge` sempre que ele não identifica uma relação adequada. Esse caso ocorreu 95 vezes, representando 20% do número total de relações. Se o analisador trocasse a relação `:null_edge` para a relação `:ARG0` (já que ela é bastante frequente), por exemplo, o resultado do analisador melhoraria 5% na medida-f. Ademais, o analisador também produz conceitos `null_tag` quando ele não identifica um conceito na sentença. Esses casos apareceram apenas duas vezes nos resultados, e são mais difíceis de se tratar. Na Figura 15, é mostrado um exemplo desses problemas. Acredita-se que essas questões surgiram devido ao *corpus* ser pequeno, uma vez que esses problemas não aparecem em *corpora* maiores do inglês.

```
(d / deitado
      :null_edge (n / null_tag
            :time (d1 / dia
                  :mod (m / meio)))
      :ARG1 (s / sol
            :mod (t / todo))
      :ARG1 (m1 / mundo)
      :null_edge (f / França))
```

Figura 15: Relação `:null_edge` e conceito `null_tag` produzidos pelo `CAMR` para a sentença "*Quando é meio dia nos Estados Unidos, o sol, todo mundo sabe, está se deitando na França.*".

O analisador `AMREager` gera conceitos `emptygraph` quando ele não identifica conceitos e relações em uma sentença. Esse problema ocorreu seis vezes nos resultados e foi um dos responsáveis pelo baixo resultado na identificação de conceitos e relações de papéis semânticos quando comparado ao analisador `CAMR`, conforme apresentado na Figura 16. Por exemplo, para a sentença "*Um dia eu vi o sol se pôr quarenta e três vezes!*", o `AMREager` gera

Analisador	Sentenças curtas		Sentenças longas	
	Smatch	SEMA	Smatch	SEMA
`RBAMR`	0,61	0,43	0,46	0,25
`RBAMR` + regras	0,62	0,44	0,48	0,27
`RBAMR` + poda	0,64	0,46	0,47	0,26
`RBAMR` + regras + poda	**0,66**	**0,48**	**0,49**	**0,28**
`CAMR`	0,51	0,40	0,40	0,22
`CAMR` + alinhamento	0,53	0,41	0,42	0,24
`CAMR` + SEMA	0,52	0,41	0,41	0,23
`CAMR` + alinhamento + SEMA	**0,54**	**0,42**	**0,44**	**0,25**
`AMREager`	0,50	0,39	0,39	0,19
`AMREager` + **alinhamento**	**0,52**	**0,40**	**0,41**	**0,21**

Tabela 4: Contribuições das adaptações.

apenas `emptygraph`. Acredita-se também que esse problema seja devido ao tamanho do *corpus*.

O analisador `NeuralAMR`, embora não produza relações `:null_edge` e nem conceitos `null_tag` e `emptygraph`, não teve boa performance. Acredita-se também que esse problema seja devido ao tamanho do *corpus*, pois modelos *seq2seq* requerem um vocabulário grande para atingirem resultados satisfatórios.

O analisador `RBAMR` obteve os melhores resultados. Esse analisador é baseado em regras e não requer alinhamento entre as palavras de uma sentença e os nós do grafo. Além disso, ele usa apenas duas ferramentas para obter informações morfossintáticas e papéis semânticos, evitando muitos erros na fase de pré-processamento.

Além dos erros destacados acima, analisaram-se erros na geração de nós (conceitos) e arestas (relações) a fim de obter mais *insights*. Como um grafo AMR possui várias relações possíveis (há mais de 100 previstas no formalismo AMR), analisar cada relação é uma tarefa laboriosa. Portanto, dividiu-se a tarefa de análise em duas subtarefas para facilitar a análise: identificação de conceitos e papéis semânticas. Para isso, usou-se a ferramenta de avaliação de Damonte et al. (2017) para visualizar a performance desses componentes.

Na Figura 16, é exibido o resultado para esses componentes. A partir dessa figura, pode-se ver que o `RBAMR` atingiu os melhores resultados em ambos os componentes. Apesar dos analisadores adaptados não terem alcançado resultados melhores do que o `RBAMR` no *corpus* analisado, acredita-se que em um *corpus* maior eles possam ter resultados superiores.

Por fim, realizou-se uma análise de etiquetas morfossintáticas identificadas automaticamente nas 143 sentenças do conjunto de teste. Nessa

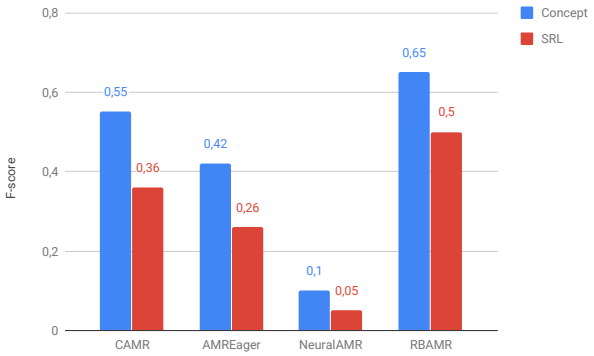

Figura 16: Resultados para identificação de conceitos e papéis semânticos.

análise, observou-se que o *parser* errou 90 etiquetas morfossintáticas, sendo a maioria em sentenças com mais de dez palavras. Isso justifica o baixo resultado em sentenças com mais de dez palavras. A etiqueta que o *parser* mais errou foi o verbo.

7. Conclusão

Neste artigo, apresentaram-se adaptações com melhorias de alguns analisadores AMR do inglês para o português. Além disso, melhorou-se o analisador AMR desenvolvido para o português através do desenvolvimento de novas regras. Essas ferramentas foram avaliadas usando o *corpus* de "O Pequeno Príncipe" anotado na língua portuguesa. Mais do que isso, apresentou-se uma análise detalhada dos resultados, mostrando as contribuições das adaptações realizadas e uma detalhada análise de erros a fim de prover *insights* para trabalhos futuros.

Embora soluções baseadas em aprendizado de máquina e, mais recentemente, abordagens baseadas em *transformers* estarem atingindo resulta-

dos impressionantes, elas requerem grandes *corpora* anotados. Para línguas com poucos recursos semânticos, como o português, grandes *corpora* ainda não são uma realidade. Nesses casos, abordagens baseadas em regras ainda têm valor significativo. Ademais, métodos adaptados podem atingir resultados interessantes, ajudando a produzir os primeiros recursos e ferramentas para uma língua.

Como trabalho futuro, pretende-se aumentar o *corpus*, adotando uma estratégia baseada em *ensemble* e *back-translation* com base nos analisadores desenvolvidos e desenvolver um método para tornar explícito o sujeito oculto.

Mais informações sobre esse trabalho e os recursos e ferramentas desenvolvidos podem ser encontrados nos portais web dos projetos OPINANDO[7] e POeTiSA[8].

Agradecimentos

Os autores agradecem ao Instituto Federal do Piauí e ao Centro de Inteligência Artificial (C4AI - http://c4ai.inova.usp.br/) da Universidade de São Paulo pelo apoio a este trabalho.

Referências

Abend, Omri & Ari Rappoport. 2013. Universal conceptual cognitive annotation (UCCA). Em *51st Annual Meeting of the Association for Computational Linguistics*, 228–238.

Anchiêta, Rafael & Thiago Pardo. 2018a. Towards AMR-BR: A SemBank for Brazilian Portuguese language. Em *11th International Conference on Language Resources and Evaluation (LREC)*, 974–979.

Anchiêta, Rafael & Thiago Pardo. 2020a. Semantically inspired AMR alignment for the Portuguese language. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1595–1600. 10.18653/v1/2020.emnlp-main.123.

Anchiêta, Rafael T., Marco Antonio Sobrevilla Cabezudo & Thiago A. S. Pardo. 2019. SEMA: an extended semantic evaluation metric for AMR. *CoRR* abs/1905.12069. arXiv.

Anchiêta, Rafael Torres & Thiago Alexandre Salgueiro Pardo. 2018b. A rule-based AMR parser for Portuguese. Em *16th Ibero-American Conference on Artificial Intelligence (IBERAMIA)*, 341–353. 10.1007/978-3-030-03928-8_28.

Anchiêta, Rafael Torres & Thiago Alexandre Salgueiro Pardo. 2020b. Exploring the potentiality of semantic features for paraphrase detection. Em *14th International Conference on Computational Processing of the Portuguese Language (PROPOR)*, 228–238. 10.1007/978-3-030-41505-1_22.

Artzi, Yoav, Kenton Lee & Luke Zettlemoyer. 2015. Broad-coverage CCG semantic parsing with AMR. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1699–1710. 10.18653/v1/D15-1198.

Ballesteros, Miguel & Yaser Al-Onaizan. 2017. AMR parsing using stack-LSTMs. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1269–1275. 10.18653/v1/D17-1130.

Banarescu, Laura, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer & Nathan Schneider. 2013. Abstract meaning representation for Sembanking. Em *7th Linguistic Annotation Workshop and Interoperability with Discourse*, 178–186.

Bender, Emily M. & Alexander Koller. 2020. Climbing towards NLU: On meaning, form, and understanding in the age of data. Em *58th Annual Meeting of the Association for Computational Linguistics*, 5185–5198. 10.18653/v1/2020.acl-main.463.

Bevilacqua, Michele, Rexhina Blloshmi & Roberto Navigli. 2021. One SPRING to rule them both: Symmetric AMR semantic parsing and generation without a complex pipeline. Em *35th AAAI Conference on Artificial Intelligence*, 12564–12573.

Blloshmi, Rexhina, Rocco Tripodi & Roberto Navigli. 2020. XL-AMR: Enabling cross-lingual AMR parsing with transfer learning techniques. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2487–2500. 10.18653/v1/2020.emnlp-main.195.

Burns, Gully A., Ulf Hermjakob & José Luis Ambite. 2016. Abstract meaning representations as linked data. Em *15th International Semantic Web Conference (ISWC)*, 12–20.

Cai, Deng & Wai Lam. 2020. AMR parsing via graph-sequence iterative inference.

[7] https://sites.google.com/icmc.usp.br/opinando/

[8] https://sites.google.com/icmc.usp.br/poetisa

Em *58th Annual Meeting of the Association for Computational Linguistics*, 1290–1301. 10.18653/v1/2020.acl-main.119.

Cai, Shu & Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. Em *51st Annual Meeting of the Association for Computational Linguistics*, 748–752.

Damonte, Marco & Shay B. Cohen. 2018. Cross-lingual abstract meaning representation parsing. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 1146–1155. 10.18653/v1/N18-1104.

Damonte, Marco, Shay B. Cohen & Giorgio Satta. 2017. An incremental parser for abstract meaning representation. Em *15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 536–546.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee & Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 4171–4186. 10.18653/v1/N19-1423.

Flanigan, Jeffrey, Sam Thomson, Jaime Carbonell, Chris Dyer & Noah A. Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. Em *52nd Annual Meeting of the Association for Computational Linguistics*, 1426–1436. 10.3115/v1/P14-1134.

Foland, William & James H. Martin. 2017. Abstract meaning representation parsing using LSTM recurrent neural networks. Em *55th Annual Meeting of the Association for Computational Linguistics*, 463–472. 10.18653/v1/P17-1043.

Goodman, James, Andreas Vlachos & Jason Naradowsky. 2016. Noise reduction and targeted exploration in imitation learning for abstract meaning representation parsing. Em *54th Annual Meeting of the Association for Computational Linguistics*, 1–11. 10.18653/v1/P16-1001.

Guo, Zhijiang & Wei Lu. 2018. Better transition-based AMR parsing with a refined search space. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1712–1722. 10.18653/v1/D18-1198.

Hardy, Hardy & Andreas Vlachos. 2018. Guided neural language generation for abstractive summarization using abstract meaning representation. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 768–773. 10.18653/v1/D18-1086.

Hartmann, Nathan, Erick Fonseca, Christopher Shulby, Marcos Treviso, Jéssica Silva & Sandra Aluísio. 2017. Portuguese word embeddings: Evaluating on word analogies and natural language tasks. Em *11th Brazilian Symposium in Information and Human Language Technology*, 122–131.

Issa, Fuad, Marco Damonte, Shay B. Cohen, Xiaohui Yan & Yi Chang. 2018. Abstract meaning representation for paraphrase detection. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 442–452. 10.18653/v1/N18-1041.

Jurafsky, Dan & James H. Martin. 2009. *Speech and language processing: An introduction to natural language processing, computational linguistics and speech recognition*. Prentice Hall.

Kingsbury, Paul & Martha Palmer. 2002. From TreeBank to PropBank. Em *3rd International Conference on Language Resources and Evaluation (LREC)*, 1989–1993.

Klein, Guillaume, Yoon Kim, Yuntian Deng, Jean Senellart & Alexander Rush. 2017. OpenNMT: Open-source toolkit for neural machine translation. Em *Proceedings of ACL 2017, System Demonstrations*, 67–72.

Konstas, Ioannis, Srinivasan Iyer, Mark Yatskar, Yejin Choi & Luke Zettlemoyer. 2017. Neural AMR: Sequence-to-sequence models for parsing and generation. Em *55th Annual Meeting of the Association for Computational Linguistics*, 146–157. 10.18653/v1/P17-1014.

Lehmann, Fritz. 1992. *Semantic networks in artificial intelligence*. Elsevier Science Inc.

Lewis, Mike, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov & Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. Em *58th Annual Meeting of the Association for Computational Linguistics*, 7871–7880. 10.18653/v1/2020.acl-main.703.

Li, Bin, Yuan Wen, Weiguang Qu, Lijun Bu & Nianwen Xue. 2016. Annotating

the little prince with Chinese AMRs. Em *10th Linguistic Annotation Workshop*, 7–15. 10.18653/v1/W16-1702.

Liu, Fei, Jeffrey Flanigan, Sam Thomson, Norman Sadeh & Noah A. Smith. 2015. Toward abstractive summarization using semantic representations. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 1077–1086. 10.3115/v1/N15-1114.

Luong, Thang, Hieu Pham & Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1412–1421. 10.18653/v1/D15-1166.

Lyu, Chunchuan & Ivan Titov. 2018. AMR parsing as graph prediction with latent alignment. Em *56th Annual Meeting of the Association for Computational Linguistics*, 397–407. 10.18653/v1/P18-1037.

Matthiessen, Christian & John A Bateman. 1991. *Text generation and systemic-functional linguistics: experiences from english and japanese*. Pinter Publishers.

Migueles-Abraira, Noelia, Rodrigo Agerri & Arantza Diaz de Ilarraza. 2018. Annotating abstract meaning representations for Spanish. Em *11th International Conference on Language Resources and Evaluation (LREC)*, 3074–3078.

Misra, Dipendra Kumar & Yoav Artzi. 2016. Neural shift-reduce CCG semantic parsing. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1775–1786. 10.18653/v1/D16-1183.

Mitra, Arindam & Chitta Baral. 2016. Addressing a question answering challenge by combining statistical methods with inductive rule learning and reasoning. Em *AAAI Conference on Artificial Intelligence*, 2779–2785. 10.1609/aaai.v30i1.10354.

Nivre, Joakim. 2004. Incrementality in deterministic dependency parsing. Em *Workshop on Incremental Parsing: Bringing Engineering and Cognition Together*, 50–57.

van Noord, Rik & Johan Bos. 2017. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *Computational Linguistics in the Netherlands Journal* 7. 93–108.

Osa, Takayuki, Joni Pajarinen, Gerhard Neumann, J Andrew Bagnell, Pieter Abbeel & Jan Peters. 2018. An algorithmic perspective on imitation learning. *Foundations and Trends in Robotics* 7(1-2). 1–179. 10.1561/2300000053.

Palmer, Martha, Daniel Gildea & Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics* 31(1). 71–106. 10.1162/0891201053630264.

Pan, Xiaoman, Taylor Cassidy, Ulf Hermjakob, Heng Ji & Kevin Knight. 2015. Unsupervised entity linking with abstract meaning representation. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 1130–1139. 10.3115/v1/N15-1119.

Peng, Xiaochang, Daniel Gildea & Giorgio Satta. 2018. AMR parsing with cache transition systems. Em *32nd AAAI Conference on Artificial Intelligence*, 4897–4904. 10.1609/aaai.v32i1.11922.

Peng, Xiaochang, Chuan Wang, Daniel Gildea & Nianwen Xue. 2017. Addressing the data sparsity issue in neural AMR parsing. Em *15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 366–375.

Pereira, Fernando CN & Stuart M Shieber. 2002. *Prolog and natural-language analysis*. Microtome Publishing.

Pourdamghani, Nima, Kevin Knight & Ulf Hermjakob. 2016. Generating English from abstract meaning representations. Em *9th International Natural Language Generation Conference*, 21–25. 10.18653/v1/W16-6603.

Qi, Peng, Yuhao Zhang, Yuhui Zhang, Jason Bolton & Christopher D. Manning. 2020. Stanza: A python natural language processing toolkit for many human languages. Em *58th Annual Meeting of the Association for Computational Linguistics*, 101–108.

Silva, João, António Branco, Sérgio Castro & Ruben Reis. 2010. Out-of-the-box robust parsing of portuguese. Em *9th International Conference on Computational Processing of the Portuguese Language (PROPOR)*, 75–85. 10.1007/978-3-642-12320-7_10.

Sobrevilla Cabezudo, Marco Antonio & Thiago Pardo. 2019. Towards a general abstract meaning representation corpus for Brazilian Portuguese. Em *13th Linguistic Annotation Workshop*, 236–244. 10.18653/v1/W19-4028.

Song, Linfeng & Daniel Gildea. 2019. SemBleu: A robust metric for AMR parsing evaluation. Em *57th Annual Meeting of the Association for Computational Linguistics*, 4547–4552. 10.18653/v1/P19-1446.

Song, Linfeng, Daniel Gildea, Yue Zhang, Zhiguo Wang & Jinsong Su. 2019. Semantic neural machine translation using AMR. *Transactions of the Association for Computational Linguistics* 7. 19–31. 10.1162/tacl_a_00252.

Song, Linfeng, Xiaochang Peng, Yue Zhang, Zhiguo Wang & Daniel Gildea. 2017. AMR-to-text generation with synchronous node replacement grammar. Em *55th Annual Meeting of the Association for Computational Linguistics*, 7–13. 10.18653/v1/P17-2002.

Song, Linfeng, Yue Zhang, Zhiguo Wang & Daniel Gildea. 2018. A graph-to-sequence model for AMR-to-text generation. Em *56th Annual Meeting of the Association for Computational Linguistics*, 1616–1626. 10.18653/v1/P18-1150.

Steedman, Mark. 1996. *Surface structure and interpretation*. MIT Press.

Steedman, Mark. 2001. *The syntactic process*. MIT Press.

Uchida, Hiroshi, Meiying Zhu & Tarcisio Della Senta. 2006. *UNL: Universal networking language*. UNDL Foundation, International Environment House.

Vanderwende, Lucy. 2015. NLPwin–an introduction. Relatório técnico. Microsoft Research tech report no. MSR-TR-2015-23.

Vanderwende, Lucy, Arul Menezes & Chris Quirk. 2015. An AMR parser for English, French, German, Spanish and Japanese and a new AMR-annotated corpus. Em *Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 26–30. 10.3115/v1/N15-3006.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser & Illia Polosukhin. 2017. Attention is all you need. Em *31st International Conference on Neural Information Processing Systems*, 5998–6008.

Vilares, David & Carlos Gómez-Rodríguez. 2018. A transition-based algorithm for unrestricted AMR parsing. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 142–149. 10.18653/v1/N18-2023.

Vinyals, Oriol, Łukasz Kaiser, Terry Koo, Slav Petrov, Ilya Sutskever & Geoffrey Hinton. 2015. Grammar as a foreign language. Em *Advances in neural information processing systems*, 2773–2781.

Wang, Chuan, Bin Li & Nianwen Xue. 2018. Transition-based Chinese AMR parsing. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 247–252. 10.18653/v1/N18-2040.

Wang, Chuan & Nianwen Xue. 2017. Getting the most out of AMR parsing. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1257–1268. 10.18653/v1/D17-1129.

Wang, Chuan, Nianwen Xue & Sameer Pradhan. 2015a. Boosting transition-based AMR parsing with refined actions and auxiliary analyzers. Em *53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 857–862. 10.3115/v1/P15-2141.

Wang, Chuan, Nianwen Xue & Sameer Pradhan. 2015b. A transition-based algorithm for AMR parsing. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 366–375. 10.3115/v1/N15-1040.

Werling, Keenon, Gabor Angeli & Christopher D. Manning. 2015. Robust subgraph generation improves abstract meaning representation parsing. Em *53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing*, 982–991. 10.3115/v1/P15-1095.

Zhang, Sheng, Xutai Ma, Kevin Duh & Benjamin Van Durme. 2019. AMR parsing as sequence-to-graph transduction. Em *57th Annual Meeting of the Association for Computational Linguistics*, 80–94. 10.18653/v1/P19-1009.

Zhou, Junsheng, Feiyu Xu, Hans Uszkoreit, Weiguang Qu, Ran Li & Yanhui Gu. 2016. AMR parsing with an incremental joint model. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 680–689. 10.18653/v1/D16-1065.

Proposta recebida em Setembro 2021 e aceite para publicação em Junho 2022.

XPTA: um *parser* AMR para o português baseado em uma abordagem entre línguas

XPTA: an AMR parser for Portuguese based on cross-lingual approach

Eloize Rossi Marques Seno
Instituto Federal de São Paulo

Helena de Medeiros Caseli
Universidade Federal de São Carlos

Marcio Lima Inácio
Universidade de São Paulo

Rafael Torres Anchiêta
Instituto Federal do Piauí

Renata Ramisch
Redação Nota 1000

Resumo

O crescente interesse pelo processamento semântico automático, especialmente por parte dos pesquisadores de Compreensão e de Geração de Língua Natural, tem levado a muitas pesquisas relacionadas ao desenvolvimento de *parsers* semânticos. E, nesse contexto, a AMR (*Abstract Meaning Representation*) é um dos formalismos de representação semântica que tem recebido mais atenção recentemente, devido à sua maneira relativamente simples de capturar o significado de uma sentença. A construção de *parsers* AMR é em grande parte baseada em córpus de referência anotados por humanos. Contudo, esse recurso é ainda bastante escasso para muitas línguas como o português. Por esse motivo, várias pesquisas têm explorado o uso de abordagens entre línguas (*cross-lingual*), que partem de córpus e *parser* existentes em uma língua fonte, para o desenvolvimento de recursos semânticos para outras línguas alvo. Dado esse contexto, este artigo descreve o XPTA, um *parser* AMR para o português (PT) que se baseia na abordagem entre línguas (*cross-lingual*, X). O XPTA parte de *parser* AMR existente para o inglês e de vários recursos linguísticos-computacionais bilíngues inglês–português e mapeia o conhecimento semântico disponível no inglês para a representação do significado equivalente em português. Uma avaliação automática do XPTA mostrou que a abordagem adotada é promissora e os valores obtidos para *Smatch* (66%, no melhor caso) apontaram que o modelo tem potencial para competir com os resultados apresentados na literatura para outros idiomas. Além da análise automática, uma análise qualitativa dos grafos gerados possibilitou identificar e categorizar os principais erros do modelo e suas possíveis causas.

Palavras chave

representação abstrata de significado, analisador semântico, abordagem entre línguas, Português

Abstract

The growing interest in automatic semantic processing, especially by researchers in Natural Language Understanding and Natural Language Generation, has lead to several researches related to the development of semantic parsers. In this context, the semantic representation formalism of AMR (Abstract Meaning Representation) has received the most attention lately, due to its relatively simple way of capturing the meaning of a sentence. The development of AMR parser is mainly based on human-produced reference corpus. However, this resource is still quite scarce for many languages such as Portuguese. For this reason, several works have explored cross-lingual approaches, which make use of corpora and parsers available for a source language, to develop semantic resources to other target languages. Given this context, this paper describes XPTA, an AMR parser for Portuguese which is based on a cross-lingual approach. XPTA makes use of an existing parser for English and several English–Portuguese bilingual resources to map the semantic knowledge available in English to equivalent meaning representation in Portuguese. An automatic evaluation of XPTA showed that the adopted approach is promising and the results obtained for Smatch (66% in the best case) suggest that the model has the potential to compete with the results presented in the literature to others idioms. In addition to the automatic analysis, a qualitative analysis of the graphs produced by the parser allowed to identify and categorize the main mistakes of the model and their possible causes.

Keywords

abstract meaning representation, semantic parser, cross-lingual approach, Portuguese

DOI: 10.21814/lm.14.1.359

1. Introdução

Uma tarefa crucial do processamento inteligente da língua natural consiste em compreender a mensagem contida em um trecho de texto (uma sentença, por exemplo) a fim de derivar conhecimento, tomar uma decisão ou produzir uma saída esperada (tradução, resposta a uma pergunta, etc.). Para tanto, nos métodos tradicionais de aprendizado de máquina, esse processo envolve a obtenção de uma representação conceitual (semântica) dos textos, capaz de abstrair as escolhas lexicais e as características morfológicas e sintáticas, e ao mesmo tempo resolver as ambiguidades. Nesse contexto, muitos formalismos de representação do significado já foram explorados como: redes semânticas (Lehmann, 1992), interlíngua UNL (*Universal Network Language*) (Uchida et al., 2006), Lógica de Primeira Ordem (Jurafsky & Martin, 2009) e, mais recentemente, a AMR (*Abstract Meaning Representation*) (Banarescu et al., 2013).

Atualmente, a AMR tem ganhado bastante popularidade na área (Damonte et al., 2017; van Noord & Bos, 2017; Peng et al., 2017; Lyu & Titov, 2018; Vilares & Gómez-Rodríguez, 2018; Anchiêta, 2020). Uma das razões se deve à sua forma relativamente simples de representar o significado das sentenças, que se baseia em grafos acíclicos dirigidos com raiz, nos quais os nós representam os conceitos e as arestas indicam as relações semânticas entre esses conceitos. Por ser uma representação simbólica, uma das vantagens da AMR é ser facilmente compreendida pelos humanos, sendo, ao mesmo tempo, de fácil manipulação pela máquina. Outra razão para a popularidade atual da AMR está na possibilidade de realizar a avaliação automática com o uso de métricas como a *Smatch* (Cai & Knight, 2013) e a SEMA (Anchiêta et al., 2019), que se baseiam em medidas clássicas de precisão, cobertura e medida-F, calculadas com base em córpus de referência.

De acordo com Bos (2016), quando comparada a outras representações formais do significado, as estruturas AMR são também mais fáceis de serem construídas, uma vez que fazem uso de recursos pré-existentes e seguem as propriedades da teoria dos grafos.

A representação AMR é baseada na estrutura argumental —papéis semânticos— dos verbos de uma sentença, geralmente fornecida pelo PropBank (Palmer et al., 2005), sendo o nó raiz representado pelo predicado principal da sentença.

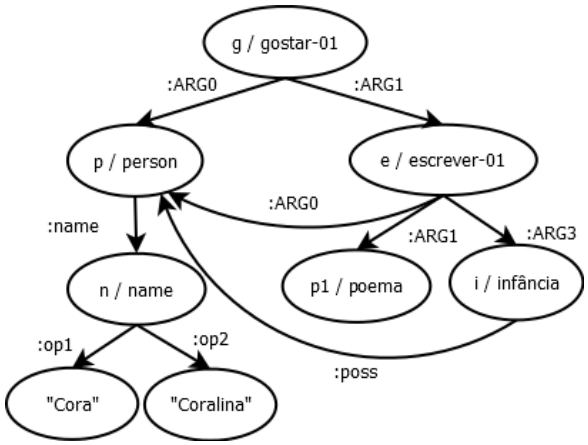

Figura 1: Grafo AMR para a sentença: *Cora Coralina gostava de escrever poemas sobre sua infância.*

Na Figura 1 é ilustrado um grafo AMR que tem como raiz o verbo "gostar" (g / gostar-01)[1]. As estruturas argumentais dos verbos em português são fornecidas pelo repositório Verbo Brasil[2] (Duran & Aluísio, 2012). Os conceitos na AMR são apresentados na sua forma canônica.

Como se pode notar na Figura 1, o predicado principal tem como argumentos o experienciador (relação indicada por :ARG0), isto é, uma entidade mencionada do tipo pessoa (p / person) com o nome (n / name) de "Cora Coralina", e o tema (relação indicada por :ARG1), representando o objeto apreciado, no caso, "escrever" (e / escrever-01)). O conceito (e / escrever-01) possui três argumentos (:ARG0, :ARG1 e :ARG3) que representam, respectivamente: o agente (isto é, o escritor (p / person)), o tema (isto é, a coisa escrita (p1 / poema)) e o assunto (i / infância). A aresta ":poss", conectando os conceitos "i / infância" e "p / person", indica um relação de posse entre eles.

O desenvolvimento de *parsers* AMR é fundamentalmente baseado em córpus de referência anotados manualmente por especialistas, uma tarefa laboriosa e demorada[3]. Devido a isso, esse tipo de recurso é, ainda, bastante escasso para boa parte das línguas. Enquanto o inglês dispõe de um córpus AMR com cerca de 59 mil sentenças[4], para o português, mais especificamente, tem-se conhecimento apenas do córpus do Pequeno Príncipe (Anchiêta & Pardo, 2018a),

[1] Cada conceito no grafo é associado a uma variável (g, p, e, n, $p1$ e i), usada para fazer referência ao conceito.

[2] `http://143.107.183.175:21380/verbobrasil/` (acessado em 29/08/2021).

[3] Banarescu et al. (2013) relatam que gastaram entre 7 e 10 minutos para anotar cada sentença do inglês.

[4] `https://amr.isi.edu/download.html` (acessado em 29/08/2021).

composto por 1.527 sentenças, do córpus jornalístico desenvolvido por Sobrevilla Cabezudo & Pardo (2019), com 299 sentenças e o OpiSums-PT-AMR (Inácio, 2021), um córpus de opiniões contendo 481 sentenças[5].

Se por um lado há carência de recursos AMR para muitas línguas, por outro lado há a disponibilidade de córpus expressivo para o inglês, o que tem proporcionado o desenvolvimento de vários *parsers* AMR (Flanigan et al., 2014; Zhou et al., 2016; van Noord & Bos, 2017; Peng et al., 2017; Lyu & Titov, 2018; Vilares & Gómez-Rodríguez, 2018), e tem ajudado a alavancar a criação desses recursos para as línguas mais carentes. Damonte & Cohen (2018), por exemplo, usaram *parser* e córpus do inglês, para o desenvolvimento de *parsers* AMR para o italiano, espanhol, alemão e chinês.

É neste contexto que o trabalho aqui descrito se insere. Mais especificamente, este artigo descreve o XPTA, um *parser* AMR para o português baseado em uma abordagem entre línguas, que parte de um *parser* AMR existente para a língua inglesa e de recursos bilíngues inglês-português e mapeia o conhecimento semântico disponível no inglês para a representação do significado equivalente em português. Para o mapeamento são utilizados diversos recursos linguísticos-computacionais bilíngues como alinhamentos lexicais e conceituais, vetores de palavras (*word embeddings*), dicionário (léxico) de tradução e repositório de verbos e estruturas argumentais, além de regras definidas com base em análise de córpus.

Os valores obtidos para *Smatch* (Cai & Knight, 2013) em nossos experimentos (58% antes da normalização e 66% após a normalização, no melhor caso) mostram que a abordagem adotada neste trabalho, embora simples, é bastante promissora, especialmente quando se observam os resultados reportados na literatura para outros idiomas, usando modelos mais sofisticados. Sheth et al. (2021), por exemplo, obtiveram um *Smatch* de 67,9% para o espanhol (melhor caso), com uma abordagem baseada em projeção de anotação, que projeta a anotação AMR disponível no inglês para as línguas alvos com base em alinhamentos de palavras contextualizadas entre sentenças paralelas. Cai et al. (2021) obtiveram um valor de 67,3% também para o espanhol (melhor caso), usando um modelo baseado em Transformadores (*Transformers*) (como será explicado na seção 3).

[5]Todos estão disponíveis em: `https://github.com/nilc-nlp/AMR-BP` (acessado em 13/09/2021).

Como principais contribuições deste trabalho destacam-se:

- O primeiro *parser* AMR para o português baseado em abordagem entre línguas: o XPTA;
- Um córpus AMR de referência para o português do domínio de divulgação científica, contendo 200 grafos;

O restante deste artigo está organizado como segue. Na Seção 2, apresenta-se, de forma breve, o formalismo de representação semântica AMR. Na Seção 3, são apresentados os principais trabalhos da literatura relacionados à este. Na Seção 4, é apresentado o *parser* AMR proposto e desenvolvido para o português, que parte de uma representação AMR do inglês e gera a equivalente em português. Na Seção 5, são apresentados o córpus de trabalho, os experimentos realizados na avaliação do modelo proposto e uma análise qualitativa dos grafos gerados pelo modelo. Por fim, na Seção 6, apresentam-se as principais conclusões deste trabalho.

2. *Abstract Meaning Representation* (AMR)

A AMR é um formalismo de representação semântica proposto por Banarescu et al. (2013). Segundo os autores, a AMR captura o significado de uma sentença abstraindo informações sintáticas como a função gramatical, as características morfossintáticas e a ordem de constituição das palavras. Essas decisões foram tomadas, de acordo com os seus criadores, de forma a facilitar e acelerar o processo de criação de bancos de anotação semântica (*sembanks*), baseando-se em recursos e ferramentas já existentes na área. Similarmente, Hovy & Lavid (2010) discutem que existe um equilíbrio necessário entre a profundidade da teoria linguística a ser utilizada e a estabilidade do processo de anotação, com isso, os autores defendem que é necessário realizar a "neutralização" ("*neutering*") da teoria linguística para tornar a anotação viável e pragmática, com o foco no objetivo que se espera alcançar – no caso da AMR, a criação de ferramentas linguístico-computacionais. Há, porém, dentro da comunidade interessada nessa representação semântica, críticos a algumas decisões feitas originalmente (por exemplo, os trabalhos de Donatelli et al. (2018) e Bonial et al. (2018)).

O significado de uma sentença em AMR é capturado a partir da sua estrutura predicado-argumentos (Palmer et al., 2005), e pode ser representado como um grafo direcionado com

raiz, no qual os nós representam os conceitos e as arestas representam as relações entre eles (Figura 1). Ao abstrair a morfologia e a sintaxe, sentenças com variações linguísticas (por exemplo, paráfrases e sinônimos) têm a mesma representação na AMR.

Os conceitos na AMR podem representar entidades, eventos, propriedades e estado, podendo ser uma palavra na sua forma lexicalizada (exemplo, "infância"), um *frameset* de verbo (exemplo, "gostar-01"), ou conceitos-chaves especiais como date-entity, percentage-entity, temporal-quantity, entre muitos outros descritos no manual da AMR[6]. As relações, por sua vez, podem ser argumentos de um verbo (:ARG0, :ARG1, etc.), relações semânticas gerais (:age, :destination, :location, :name, etc.), relações que indicam quantidades (:unit, :scale, :quant, :volume-quantity, etc.), relações para datas (:day, :month, :year, :time, :decade, etc.), relações para listas (:op1, :op2, etc.) ou relações discursivas (:cause, :purpose, :concession, :manner, etc.). No total, são fornecidas cerca de 100 relações pré-definidas.

Além da representação por meio de grafos, outras representações possíveis são a Lógica de Primeira Ordem e a anotação de PENMAN (Matthiessen & Bateman, 1991). Por exemplo, a Figura 2 apresenta a estrutura AMR para a sentença da Figura 1 na notação PENMAN.

```
(g / gostar-01
    :ARG0 (p / person
       :name (n / name
          :op1 ‘‘Cora’’
          :op2 ‘‘Coralina’’))
    :ARG1 (e / escrever-01
       :ARG0 p
       :ARG1 p1 / poema
       :ARG3 i / infância
          :poss p))
```

Figura 2: Notação PENMAN para a sentença: *Cora Coralina gostava de escrever poemas sobre sua infância.*

A AMR prevê também representações para diversos fenômenos linguísticos como correferência, modalidade, negação, cópula, relações inversas (como :ARG1-of, :location-of), conjunções, entidades mencionadas, entre outros. Exemplos de correferência podem ser observados na Figura 2, onde a variável *p* faz referência ao conceito *person*, remetendo à entidade "Cora Coralina", que representa o agente (:ARG0) de "escrever-01", e ao mesmo tempo se relaciona com o conceito (i / infância) por meio da relação ":poss", que se refere à infância de Cora.

Representações abstratas do significado como a apresentada na Figura 2 são muito úteis para diversas aplicações que processam sentenças em língua natural. Por exemplo, na literatura há aplicações da AMR na Sumarização Automática (Liu et al., 2015; Dohare & Karnick, 2017; Liao et al., 2018), na Tradução Automática (Song et al., 2019), na Extração de Informações (Garg et al., 2016) e em Sistemas de Perguntas e Respostas (Mitra & Baral, 2016).

Song et al. (2019), por exemplo, utilizam representações AMR de sentenças fontes como conhecimento adicional em um modelo de tradução neural (*Neural Machine Translation*), o paradigma de Tradução Automática considerado o estado da arte. Em experimentos reportados pelos autores, eles relatam um ganho de 2 pontos na medida BLEU (Papineni et al., 2002) para a tradução de inglês para alemão quando a AMR foi incorporada ao modelo. Trabalhos como esse mostram o impacto positivo do uso de AMR em aplicações de PLN. Contudo, para que seja possível manipular as representações semânticas das sentenças, as AMRs correspondentes precisam ser fornecidas por especialistas humanos ou geradas automaticamente por meio de um *parser*.

A seção seguinte descreve alguns trabalhos da literatura que investigam o mapeamento semântico entre línguas, o que se convenciona chamar de abordagem entre línguas (*cross-lingual approach*, no inglês).

3. Trabalhos Relacionados

Embora a AMR não tenha sido projetada para ser uma língua universal (Banarescu et al., 2013), como é o caso da interlíngua UNL (Uchida et al., 2006), ao abstrair a função morfológica, gramatical e a ordem das palavras de uma sentença, ela também abstrai diversas idiossincrasias linguísticas que representam as principais diferenças entre as línguas. Nesse sentido, a AMR se assemelha a uma interlíngua.

Xue et al. (2014) corroboram em parte essa afirmação ao verificar, por meio da comparação de grafos AMR do inglês com grafos do tcheco e do chinês, que há bastante compatibilidade estrutural, principalmente entre o inglês e o chinês. Segundo os autores, muitas das divergências encontradas decorreram de diferentes interpretações dos anotadores humanos no nível sintático e de traduções divergentes. Defendem que um refinamento nos padrões de anotação poderia resolver muitas dessas diferenças.

[6] https://github.com/amrisi/amr-guidelines/blob/master/amr.md (acessado em 29/07/2021).

	Abordagem	Alemão	Italiano	Espanhol	Chinês
Damonte & Cohen (2018)	Proj. Anot (alinh *SMT*)	39,0%	43,0%	42,0%	35,0%
Blloshmi et al. (2020)	LSTM	53,0%	58,1%	58,0%	41,5%
Cai et al. (2021)	*Transformer*	**64,0%**	65,4%	67,3%	**56,5%**
Sheth et al. (2021)	Proj. Anot (alinh contexto)	62,7%	**67,4%**	**67,9%**	-

Tabela 1: Valores para medida-F obtidos com *Smatch* reportados na literatura para outros idiomas.

Com base na suposição de que a representação AMR pode ser mapeada (e é a mesma) em qualquer idioma, diversos trabalhos recentes têm explorado as propriedades que são preservadas entre as línguas na AMR visando, principalmente, minimizar os esforços despendidos na criação de recursos linguísticos baseados em semântica para as línguas que carecem desses recursos. As abordagens propostas partem de córpus paralelos bilíngues e *parsers* AMR existentes na língua inglesa, a fim de obter a representação AMR equivalente na língua alvo.

Damonte & Cohen (2018) foram os precursores nessa área e treinaram *parsers* AMR para as línguas alvo italiano, espanhol, alemão e chinês. A proposta dos autores é baseada em um modelo de projeção de anotação que parte de córpus paralelos bilíngues e projeta a anotação AMR fornecida por um *parser* do inglês para a representação correspondente nas línguas alvo. Inicialmente, as sentenças paralelas foram alinhadas lexicalmente (daqui em diante, alinhamentos lexicais), usando um modelo de alinhamento de palavras tradicionalmente usado em *Statistical Machine Translation – SMT* (Dyer et al., 2013), e as palavras de cada sentença em inglês foram alinhadas aos respectivos conceitos no grafo AMR (daqui em diante, alinhamentos conceituais), usando o alinhador JAMR (Flanigan et al., 2014). Depois de obter os alinhamentos lexicais e conceituais, os autores projetaram o alinhamento entre cada sentença na língua alvo e o grafo AMR correspondente à sua tradução no inglês. Por fim, esses alinhamentos foram usados para o treinamento de *parsers* nas línguas alvo. O trabalho descrito neste artigo também usa alinhamentos conceituais e alinhamentos lexicais entre sentenças paralelas, a fim de mapear/traduzir a anotação AMR do inglês para o português. Porém, diferente do trabalho de Damonte & Cohen (2018), o modelo aqui proposto não requer treinamento. Ao invés disso, ele faz uso de um conjunto de regras e de diversos recursos linguísticos-computacionais bilíngues, para a tradução conceitual para o português. O XPTA é inovador no sentido de utilizar outros recursos bilíngues no mapeamento conceitual de uma AMR-fonte (em inglês) para uma AMR-alvo (em português), não sendo dependente do alinhamento lexical como em Damonte & Cohen (2018).

De forma similar a proposta por Damonte & Cohen (2018), Sheth et al. (2021) também treinaram *parsers* AMR para várias línguas alvo, a partir da projeção da anotação de grafos AMR do inglês. Contudo, a projeção da anotação AMR-fonte para a AMR-alvo é baseada no alinhamento de palavras que compartilham contextos similares, fornecidos por um modelo pré-treinado de vetores de palavras multilíngues, o XLM-RoBerta (Conneau et al., 2020), baseado em Transformadores (*Transformers*) (Vaswani et al., 2017). Os alinhamentos conceituais foram obtidos pelo JAMR (Flanigan et al., 2014) e pelo alinhador proposto por Pourdamghani et al. (2014). Com base nos alinhamentos, os autores projetaram o alinhamento entre os grafos AMR-fonte e as sentenças alvo e, posteriormente, treinaram os *parsers* nas línguas alvo, usando o modelo de *parsing* baseado em transição com Transformadores de Pilha (*Stack-Transformers*) proposto por Fernandez Astudillo et al. (2020).

Ao contrário dos trabalhos citados até o momento, a abordagem de Blloshmi et al. (2020) não usa alinhamentos. Mais especificamente, com o intuito de eliminar a necessidade dos alinhamentos, os autores empregaram um modelo de aprendizado por transferência, o seq2seq (*sequence-to-sequence*) (Zhang et al., 2019). A ideia por trás desse modelo é aproveitar a anotação AMR existente para o inglês para construir modelos de aprendizado que sejam capazes de generalizar e reproduzir esse tipo de anotação para outras línguas. A abordagem dos autores é baseada em duas etapas: identificação de conceitos e identificação de relações. A primeira etapa é realizada pelo modelo seq2seq, que usa uma rede neural BiLSTM (*Bidirectional Long-Short Term Memory*) como *encoder* e outra unidirecional LSTM como *decoder*. A partir das sentenças de entrada, o modelo produz uma lista de conceitos (ou nós). Para a identificação de relações (segunda etapa), os autores utilizaram um classificador *bi-affine* desenvolvido por Dozat & Manning (2017), que cria relações entre os nós identificados na etapa anterior, buscando sempre pela relação com maior pontuação entre as arestas possíveis.

A proposta de Cai et al. (2021) também é baseada no modelo seq2seq, porém, durante o treinamento, os autores fizeram uso de entradas bilíngues (isto é, sentenças na língua fonte concatenadas às respectivas traduções na língua alvo) e de uma tarefa auxiliar, que visava a predição da sentença original de entrada (em inglês). Segundo os autores, as entradas bilíngues, juntamente com a tarefa auxiliar, favoreceram a predição de conceitos AMR mais precisos. Outra diferença em relação ao trabalho de Blloshmi et al. (2020) está no uso de um modelo de Transformadores no *encoder-decoder* (Vaswani et al., 2017), enquanto que em Blloshmi et al. (2020) foi utilizado um modelo baseado em Redes Neurais Recorrentes (LSTM).

A Tabela 1 sumariza os resultados alcançados pelos modelos de Damonte & Cohen (2018), Blloshmi et al. (2020), Cai et al. (2021) e (Sheth et al., 2021) para medida-F calculada pela *Smatch* (conforme será explicada na seção 5.2.1). Essas quatro estratégias foram avaliadas utilizando o conjunto de teste do córpus LDC2017T10[7], que dispõe de sentenças em inglês e as respectivas traduções para o italiano, espanhol, alemão e mandarim chinês, sendo 1.371 sentenças para cada língua. Sheth et al. (2021), entretanto, fizeram essa avaliação somente para os três primeiros idiomas.

Conforme se pode observar na Tabela 1, o desempenho de cada estratégia varia de acordo com o idioma. Por exemplo, para o alemão e o chinês o modelo baseado em Transformadores de Cai et al. (2021) obteve os melhores resultados, enquanto que para o italiano e o espanhol, o modelo de projeção de anotação que usa alinhamentos contextuais (Sheth et al., 2021) obteve o melhor desempenho.

Apesar das divergências linguísticas existentes entre as línguas e, sobretudo, as diferenças estruturais e conceituais que frequentemente decorrem do processo de tradução de uma língua fonte para uma língua alvo, os trabalhos aqui mencionados mostraram que as abordagens entre línguas têm potencial para superar muitas dessas divergências.

4. O *parser* XPTA

O XPTA parte de um *parser* AMR existente na língua inglesa e, com base em recursos bilíngues inglês-português, mapeia/traduz o conhecimento semântico disponível na representação do significado da sentença na língua fonte (inglês) para a representação equivalente na língua alvo (português). Em outras palavras, trata-se de um modelo de tradução conceitual no qual os conceitos (nós) dos grafos AMR-fonte são traduzidos (mapeados) para os conceitos dos grafos do AMR-alvo. A Figura 3 traz uma ilustração das etapas implementadas no XPTA. O mapeamento é baseado em duas etapas de processamento em *pipeline*, a saber: (1) **alinhamento conceitual** entre grafos e sentenças em inglês e (2) **tradução conceitual e lexicalização** do grafo AMR-fonte para o grafo AMR-alvo.

O **alinhamento conceitual** (etapa 1) consiste em identificar as correspondências entre palavras/conceitos da sentença em inglês com um fragmento do grafo, representado por um ou mais nós. Essa etapa é bastante relevante, pois permite obter as correspondências conceituais também entre o grafo AMR-fonte e a sentença paralela na língua alvo (se esta estiver disponível), por meio do alinhamento lexical entre as sentenças paralelas.

O alinhamento conceitual é realizado pelo alinhador proposto por Flanigan et al. (2014), o qual se baseia em um conjunto de regras definidas manualmente. Nos experimentos realizados pelos autores, esse alinhador apresentou um desempenho de 92% de precisão, 89% de cobertura e 90% de medida-F.

A Figura 4 apresenta a sentença em inglês e seu respectivo grafo (notação PENMAN), ilustrado na Figura 3, no formato de entrada do alinhador conceitual. Esse arquivo de entrada é fornecido pelo *parser* AMR do inglês[8]. A saída do alinhador é um arquivo semelhante ao da entrada, acrescido dos alinhamentos entre um ou mais *tokens* da sentença e um ou mais nós do grafo, como no exemplo apresentado na Figura 5. Os seguintes alinhamentos foram gerados pelo algoritmo para o exemplo da Figura 3: "o3 / orangutan" com *orangutan* (isto é, alinhamento 5-6|0), "o2 / old" e "m / most" com *oldest* (alinhamento 4-5|0.0+0.0.0) e "t0 / tooth" com *teeth* (alinhamento 1-2|0.1).

Na **tradução conceitual e lexicalização** (segunda etapa), o alinhamento conceitual produzido na etapa 1 e os diversos recursos bilíngues são usados como âncora para a tradução conceitual e a lexicalização, que consiste na escolha da unidade lexical que melhor representa cada conceito da AMR-fonte na língua alvo. A entrada para esta etapa do processamento consiste na saída gerada pelo alinhador conceitual (con-

[7] https://catalog.ldc.upenn.edu/LDC2017T10 (acessado em 24/03/2022)

[8] Outras anotações também fornecidas pelo *parser*, como informações de *Part-of-Speech* e NER, foram omitidas na figura por questão de espaço.

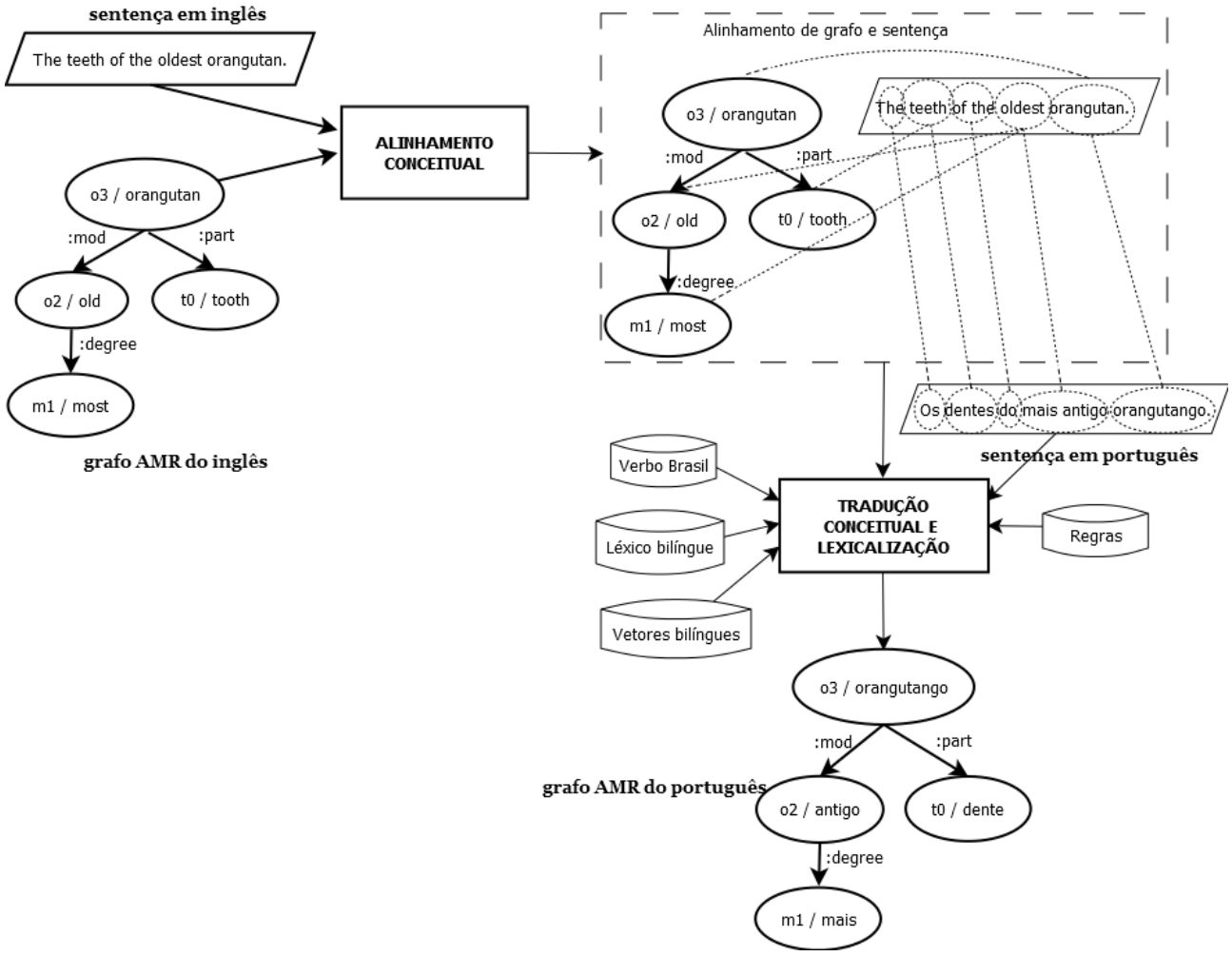

Figura 3: Etapas do XPTA, com destaque para a tradução conceitual de grafo AMR-fonte (inglês) para o AMR-alvo (português).

```
# ::snt   The teeth of the oldest orangutan
# ::tok   The teeth of the oldest orangutan
# ::node     o2     old        4-5
# ::node     m1     most       3-4
# ::node     t0     tooth      1-2
# ::node     o3     orangutan   5-6
# ::edge     old    :mod-of     orangutan     o2     o3
# ::edge     most   :degree-of  old          m1     o2
# ::edge     tooth   :part-of   orangutan     t0     o3
(o3 / orangutan
    :mod (o2 / old
      :degree (m1 / most))
    :part (t0 / tooth))
```

Figura 4: Sentença em inglês e seu respectivo grafo (notação PENMAN) no formato de entrada do alinhador conceitual.

forme Figura 5) e nos alinhamentos entre as palavras da sentença em inglês e as palavras da sentença paralela em português. Na Figura 6 é apresentado um exemplo de alinhamento lexical para as sentenças paralelas "The teeth of the oldest orangutan" e "Os dentes do mais antigo orangutango" no formato de entrada esperado pelo tradutor conceitual. No exemplo, os seguintes alinhamentos foram produzidos: "the" e "o", "teeth" e "dentes", "of" e "do", "oldest" e "mais antigo" e "orangutan" e "orangutango".

```
# ::snt   The teeth of the oldest orangutan
# ::tok   The teeth of the oldest orangutan
# ::node      o2      old 4-5
# ::node      m1      most 3-4
# ::node      t0      tooth 1-2
# ::node      o3      orangutan 5-6
# ::edge      old     :mod-of orangutan o2 o3
# ::edge      most    :degree-of old m1 o2
# ::edge      tooth   :part-of orangutan t0 o3
# ::alignments 5-6|0 1-2|0.1 4-5|0.0+0.0.0 ::annotator
Aligner v.03 ::date 2020-10-08T14:53:40.073
# ::node      0     orangutan 5-6
# ::node      0.0   old 4-5
# ::node      0.0.0 most 4-5
# ::node      0.1   tooth 1-2
# ::root      0     orangutan
# ::edge      old         degree   most    0.0 0.0.0
# ::edge      orangutan   mod      old     0     0.0
# ::edge      orangutan   part     tooth   0     0.1
(o3 / orangutan
    :mod (o2 / old
      :degree (m1 / most))
    :part (t0 / tooth))
```

Figura 5: Saída fornecida pelo alinhador conceitual para a sentença e o grafo apresentados na Figura 3.

```
os dentes do mais antigo orangotango
NULL ( ) the ( 1 ) teeth ( 2 ) of ( 3 ) the ( ) oldest ( 4 5 ) orangutan ( 6 )
```

Figura 6: Alinhamentos lexicais entre sentenças paralelas inglês-português.

A tradução conceitual e lexicalização ocorre por meio de 3 passos, nesta ordem: (1) mapeamento de verbos por meio da consulta ao Verbo Brasil, (2) mapeamento dos conceitos via aplicação das regras e (3) mapeamento dos conceitos usando os diversos recursos linguísticos-computacionais bilíngues.

Em um primeiro momento, os predicados verbais são mapeados usando os *framesets* fornecidos pelo Verbo Brasil (Duran & Aluísio, 2012), os quais estão alinhados aos *framesets* do PropBank (Palmer et al., 2005). Os verbos modais (representados na AMR por *possible-01*, *obligate-01*, *permit-01*, etc.) não são mapeados para o português, uma vez que não estão no Verbo Brasil ou não estão alinhados aos seus correspondentes no inglês. Vale dizer também que os verbos de cópula não são representados na AMR e, portanto, não aparecem no grafos AMR-fonte.

Após o mapeamento de verbos, os conceitos restantes são mapeados com base na aplicação de 5 regras que foram propostas a partir da análise do córpus:

- **REGRA 1: Entidades Nomeadas**
  Nos subgrafos com raiz representando uma entidade nomeada (por exemplo, *location*, *country*, *state*, *person*, etc.), a raiz e o seu descendente direto (isto é, o conceito *name*) são preservados e somente os descendentes de *name*, que carregam o nome da entidade e se relacionam com esse por meio de arestas do tipo *:op1*, *:op2*, etc., são mapeados para o português e colocados em ordem. A Figura 7-a traz um exemplo de aplicação desta regra no qual a entidade nomeada "Southern Hemisphere" é mapeada e lexicalizada para "Hemisfério Sul".

- **REGRA 2: Advérbios**
  Seguindo as *guidelines* da AMR[9], advérbios terminados em "ly' (por exemplo, *universally*, *incredibly*, *extremely*, etc.), cujas traduções no português terminam em "mente", são mapeados para a sua forma adjetiva no português (por exemplo, universalmente → universal, incrivelmente → incrível, extremamente → extremo, etc.). A Figura 7-b ilustra a aplicação desta regra para o advérbio "universal".

- **REGRA 3: *Frames* especiais e reificação**
  Conceitos representando *frames* especiais da AMR (*have-rel-role-91* e *have-org-role-91*) ou reificação (por exemplo, *have-purpose-91*, *be-*

[9] https://github.com/amrisi/amr-guidelines/blob/master/amr.md (acessado em 29/08/2021).

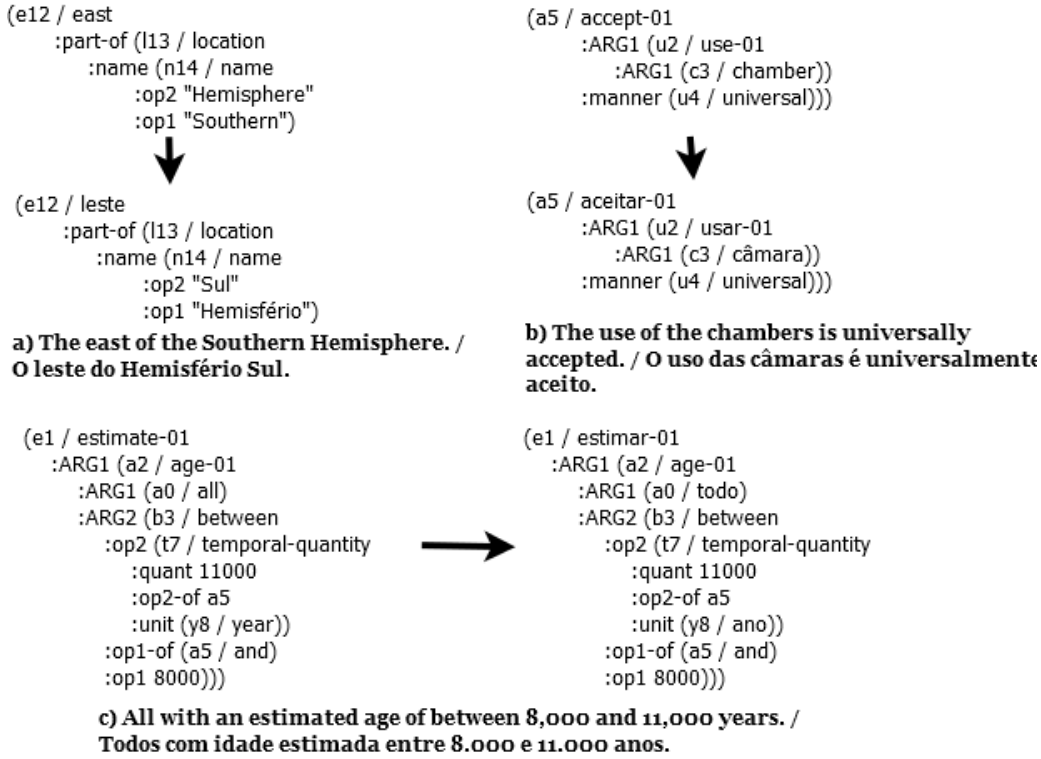

Figura 7: Exemplos de tradução conceitual de grafos AMR (formato PENMAN).

from-91, *age-01*, etc.) são preservados em inglês. A Figura 7-c traz um exemplo de mapeamento em que o conceito reificado "*age-01*" é mantido na representação AMR-alvo.

- **REGRA 4: Conceitos-chaves**
  Conceitos da AMR representando quantidades (como *monetary-quantity*, *temporal-quantity*, *distance-quantity*, etc.) ou entidades como datas (*date-entity*), porcentagens (*percentage-entity*), urls (*url-entity*), entre outras, são preservados em inglês. A Figura 7-c ilustra o mapeamento da quantidade de tempo (*temporal-quantity*) do inglês para o português, no qual apenas o conceito "year" foi alterado para "ano", preservando-se o restante da estrutura e seus valores.
- **REGRA 5: Conjunções**
  Conjunções como *and*, *or*, *contrast-01*, *either* e *between* são preservadas em inglês, como ocorre com "between" e "and" no exemplo da Figura 7-c.

O mapeamento de conceitos gerais, ou seja, para os quais nenhuma regra se aplica, é realizado com base nos recursos bilíngues. Neste trabalho, investigou-se o uso de três recursos bilíngues, aplicados nesta ordem: (1) alinhamento lexical entre a sentença fonte e a sentença alvo, (2) vetores bilíngues fonte-alvo e (3) léxico bilíngue fonte-alvo.

1. **Alinhamento lexical**
   Para o alinhamento lexical fonte-alvo, utilizou-se o alinhador GIZA++[10] (Och & Ney, 2004). Os alinhamentos lexicais permitem identificar as palavras (conceitos) correspondentes entre duas sentenças paralelas: a original (fonte) e a correspondente tradução (alvo). Um exemplo de alinhamento lexical obtido via GIZA++ pode ser visto na Figura 6. Embora bastante útil para o processo de tradução e lexicalização, nem sempre as sentenças paralelas estão disponíveis. Neste caso, os demais recursos se tornam bastante úteis.
2. **Vetores bilíngues**
   Os vetores de palavras (*word embeddings*) bilíngues usados neste trabalho são os disponibilizados pelo MUSE[11]. Esses vetores foram treinados para o par de línguas inglês-português a partir de córpus multilíngue (e não paralelo) da Wikipedia.
3. **Léxico bilíngue**
   O léxico bilíngue (dicionário de tradução) utilizado neste trabalho é o disponível no PorTAl[12] (Vieira & Caseli, 2011). Esse léxico foi gerado a partir de alinhamentos lexicais ob-

[10] https://github.com/moses-smt/giza-pp (acessado em 11/08/2021).
[11] https://github.com/facebookresearch/MUSE/blob/master/README.md (acessado em: 11/08/2021).
[12] http://www.lalic.dc.ufscar.br/portal/ (acessado em 11/08/2021).

tidos pelo GIZA++ para o córpus paralelo inglês-português da FAPESP (Aziz & Specia, 2011).

```
(o3 / orangutango
    :mod (o2 / antigo
       :degree (m1 / mais))
    :part (t0 / dente))
```

Figura 8: Grafo AMR gerado pelo XPTA para a sentença "Os dentes do mais antigo orangutango.".

Após encontrar o melhor correspondente lexical fonte para um conceito alvo, seguindo a ordem de aplicação de recursos apresentada anteriormente, cada conceito é, então, mapeado para a sua forma canônica usando o lematizador do UDPipe[13].

A saída do processo de tradução conceitual e lexicalização é o grafo AMR em português correspondente à entrada em inglês na notação PENMAN, conforme ilustrado na Figura 8.

5. Experimentos e resultados

O XPTA foi avaliado automaticamente em um conjunto de 200 sentenças. A avaliação foi realizada intrinsecamente comparando os grafos AMR produzidos pelo *parser* proposto com grafos de referência criados por anotadores humanos. Para essa avaliação, foram usadas três medidas automáticas conhecidas na área: *Smatch* (Cai & Knight, 2013), SEMA (Anchiêta et al., 2019) e SemBleu (Song & Gildea, 2019). Além da avaliação automática, os grafos gerados automaticamente foram manualmente analisados pelos anotadores, com o propósito de categorizar e contabilizar os principais erros cometidos pelo modelo.

A subseção a seguir descreve o córpus utilizado nos experimentos, bem como a construção do córpus de referência. A avaliação automática e a análise manual serão descritas nas subseções 5.2 e 5.3, respectivamente.

5.1. Preparação do córpus

Para os experimentos realizados neste trabalho foi adotado o córpus paralelo FAPESP (Aziz & Specia, 2011), composto por textos de divulgação científica escritos originalmente em português e suas respectivas traduções para o inglês, os quais foram extraídos da revista científica Pesquisa FAPESP[14].

Para as avaliações do XPTA foram usados 200 pares de sentenças paralelas, conforme será explicado na subseção 5.1.1.

Os grafos AMR correspondentes às sentenças na língua fonte (inglês, neste caso) foram obtidos pelo *parser* proposto por Lyu & Titov (2018). Baseado em um modelo neural, esse *parser* representa o estado da arte para a língua inglesa com medida-F reportada pelos autores de 74,4%.

Como a construção de córpus AMR é uma tarefa bastante laboriosa, optou-se por gerar as referências a partir da pós-edição de grafos produzidos automaticamente, reduzindo, assim, a complexidade da tarefa. Assim, para a construção do córpus AMR de referência na língua alvo, foram utilizados como ponto de partida os grafos AMR do português gerados pelo XPTA (vide seção 4).

A subseção a seguir descreve a construção do córpus de referência.

5.1.1. Construção do córpus AMR de referência

Para a construção do córpus AMR de referência na língua alvo (português), foram selecionados os 200 pares de sentenças paralelas do córpus FAPESP melhores ranqueados de acordo com o *score* de alinhamento produzido pelo Giza++ (Och & Ney, 2004). Esse critério de seleção das sentenças está fundamentado no fato do modelo de tradução conceitual ter como passo inicial o uso dos alinhamentos lexicais, de modo que erros nesse processo impactariam negativamente o desempenho do XPTA. Além do mais, as sentenças com melhores alinhamentos tendem a ser mais curtas, o que também simplifica a produção do grafo AMR-fonte.

A partir dos grafos AMR produzidos para o inglês para essas 200 sentenças, foram gerados os grafos paralelos no português, usando a configuração completa do XPTA descrita na seção 4. Posteriormente, os grafos na língua alvo (português) foram pós-editados por 4 anotadores nativos do português, todos pesquisadores da área de PLN com experiência em anotação de córpus AMR.

Com o objetivo de estabelecer um padrão de revisão e de correção dos grafos, os anotadores participaram de duas sessões de treinamento. Na primeira sessão, todos editaram conjuntamente 20 grafos[15]. Na segunda sessão, outros 20 gra-

[13] https://github.com/TakeLab/spacy-udpipe (acessado em 11/08/2021).

[14] http://revistapesquisa.fapesp.br/ (acessado em 11/08/2021).

[15] O córpus de referência e as diretrizes da anotação/pós-edição estão disponíveis em: https:

fos foram dados aos 4 anotadores para realizar a edição de forma individual, o que possibilitou calcular a concordância entre eles e aprimorar a tarefa por meio de discussões sobre os principais erros cometidos.

Após as sessões de treinamento, os 160 grafos restantes do córpus foram divididos em 4 conjuntos de 40 grafos cada, que foram pós-editados em 4 etapas. Durante cada etapa eram formadas 2 duplas de anotadores e cada dupla era responsável por revisar metade dos grafos (isto é, 20) daquele conjunto. Os dois membros de cada dupla revisavam, separadamente, os 20 grafos daquela etapa. Posteriormente, a concordância entre os membros de cada dupla era calculada. A fim de garantir maior homogeneidade na tarefa de pós-edição de grafos, as duplas eram refeitas a cada nova etapa não havendo, portanto, repetição da mesma dupla para conjuntos diferentes de grafos.

A concordância entre os anotadores foi calculada usando a medida *Smatch* (Cai & Knight, 2013), conforme será explicada na seção 5.2.1. Essa medida tem sido amplamente usada na literatura (Banarescu et al., 2013; Damonte & Cohen, 2018; Anchiêta et al., 2019), tanto no cálculo da concordância entre anotadores como na comparação de grafos AMR produzidos automaticamente com grafos de referência.

A concordância entre todos os anotadores foi calculada a partir do valor *Smatch* obtido por cada dupla de anotadores, considerando todas as combinações de pares possíveis. Durante o treinamento, a média desses valores foi de 0,77, sendo que o menor e o maior valor de *Smatch* obtidos foram 0,72 e 0,86, respectivamente. Após o treinamento, a concordância média entre as duplas na pós-edição do córpus foi de 0,78, sendo 0,70 e 0,87 o menor e o maior valor de *Smatch* alcançados, respectivamente.

Os valores de concordância obtidos neste trabalho são superiores aos reportados por outros trabalhos da literatura. Banarescu et al. (2013), por exemplo, alcançaram valores entre 0,70 e 0,80 (0,71 em média) na construção de um córpus AMR para o inglês, enquanto Sobrevilla Cabezudo & Pardo (2019) obtiveram valores de *Smatch* entre 0,70 e 0,77 (0,72 em média) na construção de um córpus AMR de notícias em português. Uma explicação possível para a alta concordância obtida entre os anotadores neste trabalho pode ser o fato de que as duplas tinham como ponto de partida da anotação o mesmo grafo gerado automaticamente pelo XPTA.

`//github.com/LALIC-UFSCar/AMRScien-Br-Corpus/` (acessado em 15/09/2021).

A *Smatch*, no entanto, não considera a equivalência entre grafos que apresentam variações de conceitos e/ou de relações, mas que são considerados equivalentes de acordo com a AMR, e atribui um baixo valor quando essas variações ocorrem. As duas estruturas AMR exibidas na Figura 9, por exemplo, são equivalentes, sendo que a (a) utiliza reificação (isto é, a relação de causa é representada por meio de um conceito), enquanto a (b) não utiliza.[16]

```
(s / sair-01
    :ARG1 (m / menino)
    :ARG1-of (c / cause-01
        :ARG0 (c1 / chegar-01
            :ARG0 (g / garoto))))
```

a) Com reificação

```
(s / sair-01
    :ARG1 (m / menino)
    :cause (c / chegar-01
        :ARG0 (g / garoto)))
```

b) Sem reificação

Figura 9: Grafos AMR equivalentes para a sentença: *A menina saiu porque o garoto chegou* (Formato PENMAN).

Com o intuito de tornar essa comparação mais justa, Goodman (2019) propôs um normalizador de grafos AMR, o Norman, que, entre outras transformações, converte um grafo não reificado para um grafo reificado, garantindo, assim, que grafos similares sejam avaliados como equivalentes. O Norman foi aplicado aos grafos pós-editados pelos anotadores e o *Smatch* foi novamente calculado. A concordância média obtida durante a fase de treinamento passou de 0,77 para 0,81, após a normalização, sendo que os valores obtidos entre cada dupla de anotadores ficaram entre 0,77 e 0,89. A concordância média obtida pelas duplas para o córpus todo (depois do treinamento), por sua vez, passou de 0,78 para 0,81, sendo o menor e o maior valor obtidos 0,74 e 0,90, respectivamente.

Exemplos de grafos de referência podem ser vistos na seção 5.3, nas Figuras 13 e 14.

5.2. Avaliação Automática

Com o objetivo de medir a qualidade dos grafos do português, produzidos a partir do mapeamento de grafos de sentenças paralelas em inglês, esses foram comparados automaticamente com os grafos de referência, gerados conforme descrito na subseção 5.1.1. A subseção seguinte descreve as

[16] Reificação é quando uma relação AMR é representada por meio de um conceito. Esse recurso é usado quando se deseja enfatizar a importância de uma relação. Por exemplo, a relação ":cause" ao ser substituída pelo conceito "cause-01" é enfatizada no grafo. Para uma explicação mais detalhada, sugere-se consultar as especificações da AMR em `https://github.com/amrisi/amr-guidelines/blob/master/amr.md` (acessado em 15/08/2021).

medidas de avaliação usadas nessa comparação. Os resultados da avaliação, por sua vez, são apresentados e discutidos na subseção 5.2.3.

5.2.1. Medidas de Avaliação

Três medidas de avaliação automática de grafos AMR disponíveis na literatura foram usadas neste trabalho, a saber: *Smatch* (Cai & Knight, 2013), SEMA (Anchiêta et al., 2019) e SemBleu (Song & Gildea, 2019).

A *Smatch* calcula a Precisão, a Cobertura e a medida-F de conceitos e relações comuns entre dois grafos AMR. Mais especificamente, cada grafo é representado como uma conjunção de proposições lógicas, nomeadas de triplas, e a Precisão (P) e a Cobertura (C) são calculadas sobre essas triplas, seguindo as Equações 1 e 2, respectivamente. Nessas equações, M representa o total de triplas corretas segundo algum grafo AMR de referência, N representa o número total de triplas produzidas por outro anotador humano ou por um *parser* e T representa o total de triplas do grafo AMR de referência. A medida-F, por sua vez, representa a média harmônica entre a Precisão e a Cobertura (Equação 3).

$$P = \frac{M}{N} \tag{1}$$

$$C = \frac{M}{T} \tag{2}$$

$$F = \frac{2 \times P \times C}{P + C} \tag{3}$$

Ao fazer a comparação entre duas representações AMR, a *Smatch* cria uma relação "TOP" para cada grafo, indicando o nó que representa a raiz. Ao avaliar, por exemplo, o grafo (a) da Figura 9 (teste) comparando-o com o grafo (b) (referência), reproduzidos na Figura 10 em formato de diagrama, a *Smatch*[17] retorna os seguintes valores:

- $M = 7$ uma vez que há 7 triplas corretas: "sair-01", "menino", "chegar-01", "garoto", "TOP", ":ARG1" entre "sair-01" e "menino" e ":ARG0" entre "chegar-01" e "garoto";
- $N = 10$ uma vez que há 10 triplas no total: "sair-01", "menino", "cause-01", "chegar-01", "garoto", "TOP", ":ARG1" entre "sair-01" e "menino", ":ARG1" entre "cause-01" e "sair-01", ":ARG0" entre "cause-01" e "chegar-01" e ":ARG0" entre "chegar-01" e "garoto") e
- $T = 8$ uma vez que há 8 triplas no grafo AMR de referência: "sair-01", "menino", "chegar-01", "garoto", "TOP", ":ARG1" entre "sair-01" e "menino", ":cause" entre "sair-01" e "chegar" e ":ARG0" entre "chegar-01" e "garoto".

Portanto, P é igual a 70% ($7 \div 10$), C é igual a 88% ($7 \div 8$) e F é igual a 78%.

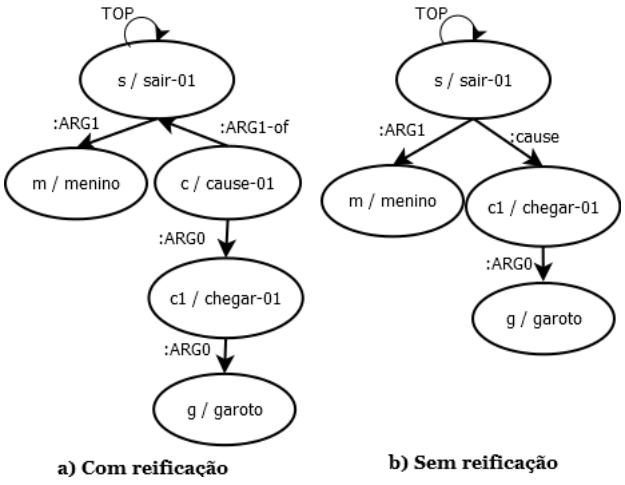

Figura 10: Grafos AMR equivalentes para a sentença: *A menina saiu porque o garoto chegou.*

A SEMA é uma extensão da *Smatch* que, ao contrário desta, leva em consideração a dependência entre os nós dos dois grafos no cálculo da Precisão e da Cobertura (Equações 1 e 2). Em outras palavras, a SEMA considera o antecedente de um nó ao fazer a comparação de triplas, enquanto *Smatch* considera apenas a contagem de triplas em comum entre dois grafos. Outra diferença é que a SEMA não inclui a relação "TOP" na raiz do grafo. Ao comparar os grafos da Figura 10 usando a SEMA tem-se P igual a 67% (isto é, $6 \div 9$, desconsiderando a relação "TOP"), C igual a 86% (ou seja, $6 \div 7$ ao desconsiderar "TOP") e F igual a 75%.

SemBleu, por sua vez, é uma variante da BLEU (Papineni et al., 2002), uma medida amplamente usada na avaliação de tradutores automáticos, que calcula a correlação entre um texto (ou sentença) traduzido automaticamente e um texto de referência produzido por um tradutor humano. De maneira similar, SemBleu calcula a correlação entre um grafo AMR produzido automaticamente (a) e outro grafo de referência (r). Diferentemente de *Smatch* e SEMA, SemBleu considera a coocorrência de n-gramas (*strings* idênticas) entre dois grafos, ao invés de usar triplas lógicas.

[17] https://www.isi.edu/cgi-bin/div3/mt/text_shu_v03.cgi (acessado em 15/08/2021).

A medida SemBleu estende a BLEU redefinindo o coeficiente de penalidade, *Brevity Penalty - BP*, que multiplica a média geométrica dos valores de precisão (pn) obtidos para cada n-grama, conforme mostrado na Equação 4. *BP* representa o tamanho de um grafo, dado pela soma do número de nós e de arestas que ele possui. A precisão (pn) de cada n-grama é dada pelo total de n-gramas em comum entre (a) e (r), dividido pelo número de n-gramas de (a). pn é ponderado por um peso positivo $wn = 1/3$, onde 3 é o comprimento máximo de N. Em outras palavras, SemBleu considera apenas uni-gramas (nós), bigramas (pares de nós diretamente conectados) e trigramas (três nós diretamente conectados). Por exemplo, para o grafo b da Figura 10 tem-se como unigramas os nós "sair-01", "menino", "chegar-01" e "garoto", como bigramas tem-se "sair-01 :ARG1 menino", "sair-01 :cause chegar-01" e "chegar-01 :ARG0 garoto" e, por fim, como trigrama tem-se "sair-01 :cause chegar-01 :ARG0 garoto".

$$SemBleu = BP \cdot \exp \sum_{n=1}^{N} w_n \log p_n \qquad (4)$$

Na comparação entre os dois grafos da Figura 10, SemBleu é igual a 46%.

5.2.2. Configurações do modelo

Com o objetivo de avaliar a contribuição individual de cada recurso linguístico-computacional usado no mapeamento dos grafos AMR-fonte para o AMR-alvo, o XPTA foi executado usando diferentes configurações que são apresentadas a seguir:

VB + ali + vet + lex Com todos os recursos, isto é, Verbo Brasil (VB), alinhamentos lexicais e conceituais, vetores e léxico bilíngues;

VB + ali + lex Usando o VB, os alinhamentos lexicais e conceituais e o léxico bilíngue;

VB + ali + vet Usando o VB, os alinhamentos lexicais e conceituais e os vetores bilíngues;

VB + vet + lex Usando o VB, léxico e vetores bilíngues;

VB + lex Usando somente o VB e o léxico bilíngue;

VB + vet Usando apenas o VB e os vetores bilíngues;

VB + ali Usando somente o VB e os alinhamentos lexicais e conceituais.

Como o Verbo Brasil é o único recurso usado no mapeamento de verbos, ele foi mantido em todas as configurações.

5.2.3. Resultados da Avaliação Automática e Discussão

As Tabelas 2 e 3 apresentam os valores obtidos por cada configuração do XPTA para *Smatch*, SEMA e SemBleu, antes e depois da normalização dos grafos pelo Norman (Goodman, 2019), respectivamente.

Com base nos valores das três medidas de avaliação usadas é possível notar um melhora substancial dos valores apresentados na Tabela 2 após a normalização (conforme Tabela 3), independente da configuração do modelo. A melhora nos valores apresentados por *Smatch* e SemBleu foi em média de 8 pontos. Para SEMA, o ganho depois da normalização foi ainda maior: de 10 a 13 pontos, dependendo da configuração. Esses resultados apontam a dificuldade dessas medidas em lidar com variações de significado equivalentes presentes nas representações AMR, e ressaltam a importância da normalização para possibilitar uma avaliação mais justa do modelo.

De maneira geral, quando se compara as diferentes configurações testadas, observa-se pouca diferença nos valores apresentados por cada uma delas, independente da medida utilizada. Os piores valores, contudo, foram alcançados quando se usou apenas o Verbo Brasil (VB) e os alinhamentos (VB + ali). De fato, como se pode observar nos resultados obtidos pela configuração que usou todos os recursos menos os alinhamentos (VB + lex + vet) e também nos resultados apresentados pelas três últimas configurações (vide Tabela 2), o léxico e os vetores bilíngues parecem ter tido um papel mais importante no mapeamento conceitual do que os alinhamentos. Uma possível explicação para isso pode ser erros decorrentes do alinhamento lexical, fornecidos pelo Giza++, ou conceitual, fornecidos pelo JAMR.

Comparando o uso do léxico e dos vetores bilíngues, nota-se que ao aplicar o VB e os alinhamentos juntamente com o léxico (VB + ali + lex), o resultado foi ligeiramente superior ao obtido quando se aplicou o VB e os alinhamentos com os vetores (VB + ali + vet), para todas as medidas. O mesmo se observa quando se exclui os alinhamentos dessas duas configurações, isto é, quando se aplica apenas o VB e o léxico (VB + lex), nota-se uma pequena melhora (de 1 até 3 pontos dependendo da medida) em relação ao uso do VB com os vetores (VB + vet). Neste caso é relevante destacar que enquanto os léxicos

Modelo	Smatch			SEMA			SemBleu F-*score*
	P	**C**	**F-*score***	**P**	**C**	**F-*score***	
VB + ali + vet + lex	58%	57%	57%	**35%**	**34%**	**35%**	**26%**
VB + ali + lex	**59%**	**57%**	**58%**	**35%**	**34%**	**35%**	**26%**
VB + ali + vet	58%	57%	57%	**35%**	**34%**	**35%**	25%
VB + lex + vet	57%	56%	57%	34%	33%	33%	25%
VB + lex	57%	56%	57%	34%	33%	33%	25%
VB + vet	55%	54%	55%	32%	32%	32%	24%
VB + ali	55%	54%	54%	32%	31%	31%	24%

Tabela 2: Valores obtidos por cada configuração do modelo para *Smatch*, SEMA e SemBleu antes da normalização.

Modelo	Smatch			SEMA			SemBleu F-*score*
	P	**C**	**F-*score***	**P**	**C**	**F-*score***	
VB + ali + vet + lex	67%	64%	65%	47%	45%	46%	33%
VB + ali + lex	**67%**	**64%**	**66%**	**48%**	**46%**	**47%**	**34%**
VB + ali + vet	67%	64%	65%	47%	45%	46%	33%
VB + lex + vet	66%	64%	65%	47%	45%	46%	**34%**
VB + lex	66%	63%	65%	47%	45%	46%	**34%**
VB + vet	65%	62%	64%	44%	42%	43%	33%
VB + ali	64%	62%	63%	43%	41%	42%	31%

Tabela 3: Valores obtidos por cada configuração do modelo para *Smatch*, SEMA e SemBleu após a normalização pelo Norman.

foram gerados para o mesmo córpus (córpus FAPESP), os vetores bilíngues são de domínio geral e foram gerados a partir de textos da Wikipedia. Entretanto, não é possível afirmar que essas diferenças são significativas, devido à falta de testes estatísticos.

É importante notar também que a configuração VB + lex mostrou-se equivalente à configuração que usou todos os recursos, ressaltando a contribuição do léxico bilíngue para o mapeamento conceitual. Vale ressaltar que o bom desempenho apresentado por recursos já disponíveis, como o léxico bilíngue do PorTAl e os vetores bilíngues do MUSE, demonstra a aplicabilidade do XPTA mesmo na ausência de uma versão paralela da sentença a ser processada.

Por fim, ressalta-se que a experimentação com diferentes configurações de recursos usados pelo XPTA foi realizada apenas a título de análise empírica do impacto de cada recurso no valor da medida de avaliação. Como o intuito não é estabelecer qual é a melhor configuração, uma vez que os recursos já existem e se complementam, nenhuma análise de significância estatística foi realizada.

Na comparação dos resultados apresentados pelas três medidas de avaliação, para todas as configurações do XPTA, os maiores valores foram obtidos com a *Smatch*. Por se tratar de uma medida mais simples e que não considera a dependência entre os nós de um grafo AMR, esse resultado já era esperado de certa forma. Isso explica também a diminuição nos valores apresentados por SEMA, quando comparados aos obtidos com a *Smatch*, que, ao contrário dessa medida, considera a dependência entre os elementos do grafo ao fazer a comparação de triplas. Por sua vez, os menores valores foram apresentados por SemBleu, que tende a ser ainda mais rígida do que SEMA, ao comparar n-gramas.

Os resultados relatados com base apenas nas medidas de avaliação automática indicam que a abordagem adotada é factível e as representações AMR podem ser compartilhadas com êxito entre as línguas, apesar de suas diferenças linguísticas e estruturais. Contudo, para permitir uma análise mais detalhada do impacto dessas diferenças na representação AMR entre idiomas diferentes, realizou-se também uma análise qualitativa dos grafos AMR em português gerados pelo XPTA.

5.3. Análise Qualitativa

Com o propósito de identificar e categorizar os principais erros cometidos pelo XPTA, durante a construção do córpus AMR de referência do português (vide seção 5.1.1), os anotadores foram solicitados a registrar cada edição (correção) que

faziam nos grafos. Mais especificamente, cada anotador recebia um arquivo com os grafos a serem pós-editados e as respectivas sentenças em português. Para cada grafo os anotadores registravam:

- o total de *frameset* de verbos substituídos, ou seja, quando outro *frameset* disponível no Verbo Brasil era substituído pelo escolhido pelo modelo;
- o total de conceitos substituídos (excluindo-se os verbos), isto é, quando o anotador fazia outra escolha lexical para representar um conceito;
- o total de relações (arestas) substituídas, já que em muitos casos a mudança de *frameset* provocava também alterações nas relações que representavam seus argumentos;
- o total de *links* (conexões) alterados, isto é, quando um nó era conectado a outro antecedente diferente do escolhido pelo modelo;
- o total de conceitos novos incluídos (inclusive verbos), isto é, que não haviam sido representados no grafo antes;
- o total de relações incluídas uma vez que, normalmente, a inclusão de um novo conceito leva à inclusão de uma relação;
- o total de conceitos removidos, inclusive verbos;
- o total de relações excluídas já que a remoção de um conceito normalmente leva à remoção de uma relação também;
- o total de conceitos lematizados, ou seja, que apresentavam erros de lematização e eram corrigidos pelos anotadores.

O gráfico da Figura 11 resume o total de edições realizadas para cada uma dessas categorias. Foram realizadas 992 edições em todo o corpus. A porcentagem de grafos em que cada tipo de edição ocorreu é apresentada na Figura 12.

Como se pode observar na Figura 11, a substituição de relações foi a principal edição realizada pelos anotadores, seguida pela substituição de conceitos gerais (não incluindo verbos). A substituição de relações e de conceitos correspondem, respectivamente, a 19% e 18% das edições realizadas. Ambas ocorreram em 67,7% dos grafos (Figura 12). Contudo, quando se considera a substituição de conceitos e de verbos (*framesets*) juntas, elas representam 25,2% de todas as correções.

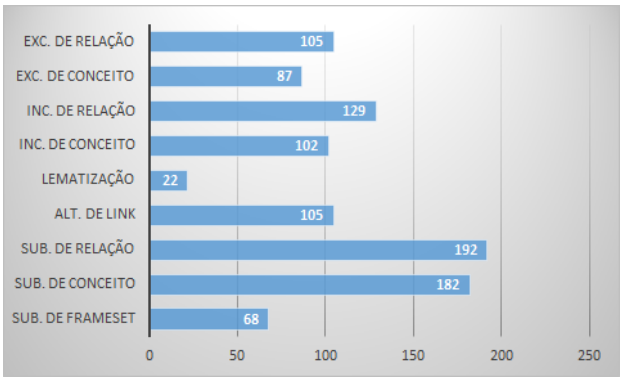

Figura 11: Total de edições realizadas por categoria.

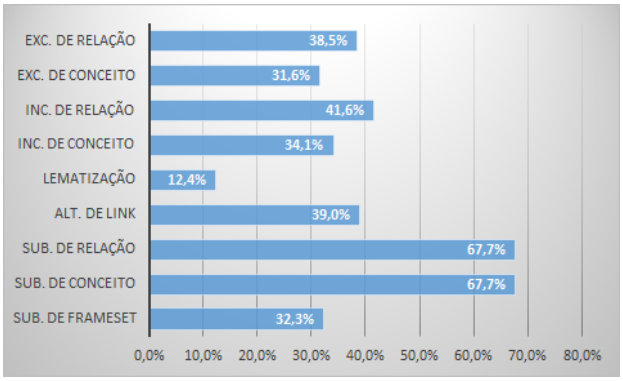

Figura 12: Porcentagem de grafos em que cada tipo de edição foi realizada.

A inclusão de novas relações também foi comum, ocorrendo em 41,6% dos grafos. Ao contrário do que se poderia esperar, a inserção de uma relação nem sempre está relacionada à inclusão de um novo conceito, conforme se observa nos gráficos das Figuras 11 e 12. A correferência é um exemplo típico de inclusão de uma relação, que não envolve a adição de um novo conceito.

A exclusão de relações e as alteração de conexões (*links*) representam, cada uma, 10,5% do total de edições. A primeira acontece quando há alguma remoção de conceito (nó) ou quando um nó deixa de ser interno e passa a ser a raiz do grafo. As alterações de conexões, por sua vez, estão associadas às mudanças na estrutura dos grafos.

Com menos frequência observa-se alterações de *framesets* de verbos (representando 6,8% do total de edições) e lematização de conceitos (representando 2,2%).

O número médio de edições realizadas pelos anotadores, em cada grafo, foi de 6. Contudo, 34,7% dos grafos tinham entre 0 e 3 erros apenas, sendo que desses 10,0% estavam corretos (nenhum erro), 7,4% com 1 erro, 9,3% com 2 erros somente e 8,0% continham exatamente 3 erros.

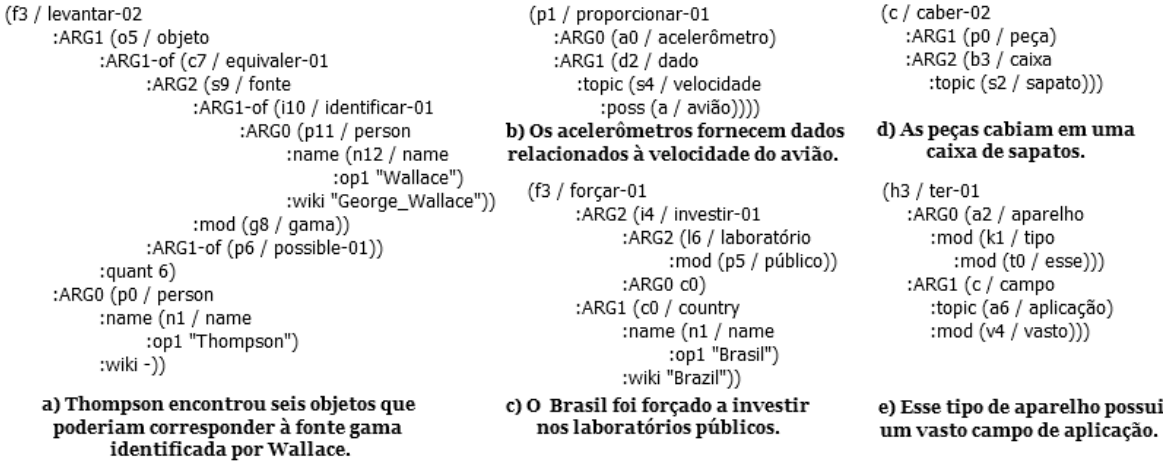

Figura 13: Exemplos de grafos corretos gerados pelo XPTA (formato PENMAN).

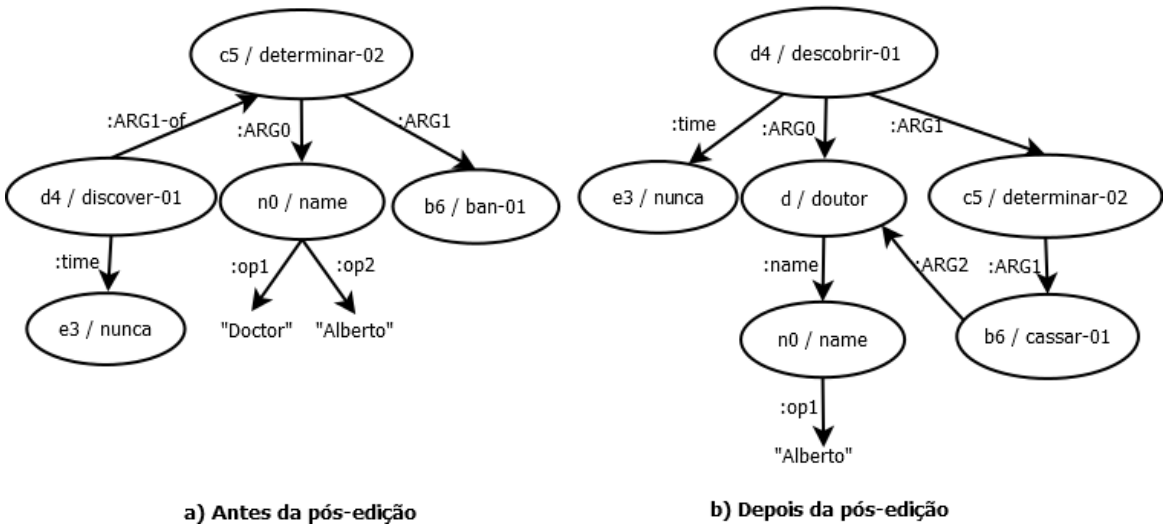

Figura 14: Exemplo de grafo gerado pelo XPTA antes e depois da pós-edição pelos humanos.

A Figura 13 apresenta 5 exemplos de grafos gerados pelo XPTA (seção 4), julgados corretos pelos anotadores. É importante dizer que o mapeamento de relações ":wiki" não foi abordado na versão atual do XPTA e, portanto, quando presentes nos grafos gerados a partir do inglês, essas relações eram simplesmente ignoradas pelos anotadores[18].

Um exemplo de grafo antes e depois da pós-edição dos anotadores pode ser visto na Figura 14. No exemplo ocorreram duas substituições de *framesets* ("discover-01" → "descobrir-01" e "ban-01" → "cassar-01"), uma troca de conceito ("Doctor" → "Alberto"), uma exclusão de conceito ("Alberto") e, consequentemente, da relação que este estabelecia com o seu antecedente (ou seja, ":op2"), uma inclusão de conceito ("doutor"), duas inclusões de relação (":name" e a correferência indicada por ":ARG2" entre "cassar-01" e "doutor") e, por fim, uma mudança de conexão (*link*) do nó "name", que tinha como antecedente "determinar-02" e passou a conectar-se ao nó "doutor", por meio da relação ":name".

Os erros observados nos grafos gerados pelo XPTA têm várias origens distintas, uma vez que cada recurso linguístico-computacional adotado pode causar algum tipo de erro no processo. Por exemplo, as falhas relacionadas aos *framesets*

[18] A relação ":wiki" é usada pela AMR para representar a forma canônica de entidades nomeadas, por exemplo, "NY" é representada por "New York City". O *parser* AMR (Lyu & Titov, 2018) usa a Wikipédia do inglês para recuperar a forma canônica de entidades nomeadas. Quando ele não encontra, a relação é representada por ":wiki -".

estão normalmente associadas à falta de um correspondente no Verbo Brasil ou, ainda, à falta de um mapeamento entre o *frameset* do inglês e o seu correspondente no português. Em outras palavras, embora haja um correspondente de um *frameset* qualquer do inglês no Verbo Brasil, este nem sempre está associado ao *frameset* do PropBank. Esse é o caso de "discover-01" e "ban-01" (Figura 14), que apesar de terem um correspondente no Verbo Brasil, isto é, "descobrir-01" e "cassar-01", respectivamente, falta uma associação entre eles. Devido à isso, não foi possível fazer o mapeamento automaticamente. Essa dependência do PropBank, no caso do inglês, e do Verbo Brasil, no caso do português, se impõe como uma das limitações da própria AMR.

Há também erros advindos do próprio grafo AMR-fonte, que foram derivados do processo de construção do mesmo pelo *parser* ou, ainda, de alguma ferramenta de pré-processamento usada por ele. Esse deve ser o caso, por exemplo, do conceito "Doctor" (na Figura 14-a), que foi incluído no grafo como um componente de uma entidade nomeada. Outro exemplo é a relação de correferência não identificada pelo *parser* do inglês, ou seja, entre "cassar-01" e a subárvore que tem com raiz "doutor".

Outro tipo de erro propagado no modelo proposto é derivado do lematizador do UDPipe. Conforme mostrado na Figura 12, os erros de lematização representam 12,4% dos erros encontrados no córpus de avaliação.

Por fim, os erros nos alinhamentos lexicais e/ou conceituais também podem causar ruídos, especialmente quando se utiliza apenas os alinhamentos no processo de mapeamento conceitual. Porém, ao combinar outros recursos como léxico e vetores bilíngues, os conceitos e palavras não alinhados geralmente são mapeados com base nesses recursos.

Embora os resultados com as medidas de avaliação automática tenham sido promissores, a partir do que foi exposto com a discussão da análise manual apresentada nesta seção, concluise que há, ainda, muito espaço para se aprimorar os resultados apresentados neste trabalho, conforme será discutido na seção 6.

6. Conclusões

Este artigo descreveu o processo de construção de um *parser* AMR para o português baseado em uma abordagem entre línguas: o XPTA. O XPTA parte de *parser* AMR existente para o inglês e de recursos linguísticos-computacionais bilíngues, além de regras definidas a partir de análise de córpus, e mapeia (traduz) o conhecimento semântico disponível na língua fonte (inglês) para gerar uma representação AMR equivalente na língua alvo (português). Alguns recursos como vetores bilíngues, léxico bilíngue e alinhamentos lexicais e conceituais podem ser empregados de forma independente pela abordagem ou conjuntamente (conforme visto na seção 5.2.2).

Enquanto algumas abordagens da literatura são completamente dependentes de córpus paralelos bilíngues (vide, por exemplo, Damonte & Cohen (2018) e Sheth et al. (2021)), uma avaliação automática do XPTA usando diferentes configurações dos recursos empregados pelo modelo mostrou a sua aplicabilidade mesmo na ausência da versão paralela da sentença a ser processada. Mais especificamente, ao adotar apenas o repositório Verbo Brasil e os vetores de palavras multilíngues como recursos de base para o mapeamento conceitual da anotação AMR do inglês para o português, o modelo atingiu um *Smatch* de 64%. Todavia, o maior valor para *Smatch*, ou seja, 66%, foi obtido usando alinhamentos lexicais entre sentenças paralelas juntamente com o Verbo Brasil e um léxico bilíngue. Como as sentenças usadas na avaliação foram selecionadas tendo como critério os melhores *scores* de alinhamento fornecidos pelo Giza++, é possível que as traduções sejam bastante fiéis às sentenças originais. Investigações futuras serão necessárias para verificar o impacto de se utilizar traduções divergentes nas configurações do modelo que usam os alinhamentos paralelos como recurso para o mapeamento conceitual.

Embora não seja possível fazer uma comparação direta com outros trabalhos da literatura, ao projetar os grafos AMR para o italiano, o espanhol, o alemão e o chinês, a partir dos grafos AMR de sentenças paralelas do inglês usando somente os alinhamentos lexicais e conceituais, Damonte & Cohen (2018) alcançaram um *Smatch* de 43% para o italiano, 42% para o espanhol, 39% para o alemão e 35% para o chinês, ao comparar os grafos produzidos pelo modelo com grafos de referência.[19] Ao avaliar a aplicabilidade do modelo de Damonte & Cohen (2018) para o português do Brasil, usando o córpus do Pequeno Príncipe (Anchiêta & Pardo, 2018a), Anchiêta & Pardo (2018b) observaram um *Smatch* de apenas 37%.

[19] Os autores relatam apenas resultados para os grafos sem normalização.

A abordagem adotada aqui é passível de ser reproduzida para outras línguas alvo, desde que disponham dos recursos e ferramentas linguístico-computacionais necessários (conforme seção 4).

Além da avaliação automática, uma análise manual dos grafos produzidos pelo XPTA permitiu categorizar os erros cometidos e identificar suas origens. A análise mostrou, por exemplo, que os erros relacionados à escolha das relações que conectam dois conceitos (192 no total, conf. Figura 11) e também relacionados à estruturação dos grafos propriamente dita (105 erros conf. Figura 11) estão entre os principais problemas identificados. Esses dois tipos de erros juntos representam cerca de 30% de todos os problemas encontrados no corpus, ou seja, 297 erros de um total de 992, de acordo com a Figura 11. Esses erros são advindos do *parser* AMR do inglês, que forneceu os grafos AMR-fonte. É importante dizer que o XPTA aborda apenas o mapeamento conceitual, preservando as relações e a estrutura dos grafos AMR-fonte.

Como investigações futuras, elencam-se duas que trariam maior benefício para o modelo aqui apresentado: (1) investigar o mapeamento de relações e (2) melhorar a associação entre os *framesets* do Verbo Brasil e o do PropBank, incluindo o mapeamento entre *frames* equivalentes já existentes no repositório. Além dessas, outra possibilidade seria investigar o mapeamento de conceitos e relações "wiki", não abordados na versão atual do XPTA.

Como continuação deste trabalho, está em desenvolvimento uma interface *web* que permitirá a comunidade de PLN o acesso rápido e gratuito ao XPTA.

Referências

Anchiêta, Rafael & Thiago Pardo. 2018a. Towards AMR-BR: A SemBank for Brazilian Portuguese language. Em *11th International Conference on Language Resources and Evaluation (LREC)*, 974–979.

Anchiêta, Rafael T. 2020. *Abstract meaning representation parsing for the Brazilian Portuguese language*: Universidade de São Paulo — ICMC. Tese de Doutoramento.

Anchiêta, Rafael T., Marco Antonio Sobrevilla Cabezudo & Thiago A. S. Pardo. 2019. SEMA: an extended semantic evaluation metric for AMR. *CoRR* abs/1905.12069. arXiv.

Anchiêta, Rafael Torres & Thiago Alexandre Salgueiro Pardo. 2018b. A rule-based AMR parser for Portuguese. Em *16th Ibero-American Conference on Artificial Intelligence (IBERAMIA)*, 341–353. 10.1007/978-3-030-03928-8_28.

Aziz, Wilker & Lúcia Specia. 2011. Fully automatic compilation of a Portuguese–English parallel corpus for statistical machine translation. Em *8th Brazilian Symposium in Information and Human Language Technology (STIL)*, 234–238.

Banarescu, Laura, Claire Bonial, Shu Cai, Madalina Georgescu, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Philipp Koehn, Martha Palmer & Nathan Schneider. 2013. Abstract meaning representation for Sembanking. Em *7th Linguistic Annotation Workshop and Interoperability with Discourse*, 178–186.

Blloshmi, Rexhina, Rocco Tripodi & Roberto Navigli. 2020. XL-AMR: Enabling cross-lingual AMR parsing with transfer learning techniques. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2487–2500. 10.18653/v1/2020.emnlp-main.195.

Bonial, Claire, Bianca Badarau, Kira Griffitt, Ulf Hermjakob, Kevin Knight, Tim O'Gorman, Martha Palmer & Nathan Schneider. 2018. Abstract meaning representation of constructions: The more we include, the better the representation. Em *11th International Conference on Language Resources and Evaluation (LREC)*, em linha.

Bos, Johan. 2016. Squib: Expressive power of Abstract Meaning Representations. *Computational Linguistics* 42(3). 527–535.

Cai, Shu & Kevin Knight. 2013. Smatch: an evaluation metric for semantic feature structures. Em *51st Annual Meeting of the Association for Computational Linguistics*, 748–752.

Cai, Yitao, Zhe Lin & Xiaojun Wan. 2021. Making better use of bilingual information for cross-lingual AMR parsing. Em *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, 1537–1547. 10.18653/v1/2021.findings-acl.134.

Conneau, Alexis, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer & Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. Em *58th Annual Meeting of the Association for Computational Linguistics*, 8440–8451. 10.18653/v1/2020.acl-main.747.

Damonte, Marco & Shay B. Cohen. 2018. Cross-lingual abstract meaning representation parsing. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 1146–1155. 10.18653/v1/N18-1104.

Damonte, Marco, Shay B. Cohen & Giorgio Satta. 2017. An incremental parser for abstract meaning representation. Em *15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 536–546.

Dohare, Shibhansh & Harish Karnick. 2017. Text summarization using abstract meaning representation. *CoRR* abs/1706.01678. arXiv.

Donatelli, Lucia, Michael Regan, William Croft & Nathan Schneider. 2018. Annotation of tense and aspect semantics for sentential AMR. Em *Joint Workshop on Linguistic Annotation, Multiword Expressions and Constructions*, 96–108.

Dozat, Timothy & Christopher D. Manning. 2017. Deep biaffine attention for neural dependency parsing. Em *5th International Conference on Learning Representations*, 24–26.

Duran, Magali Sanches & Sandra M. Aluísio. 2012. Propbank-Br: a Brazilian treebank annotated with semantic role labels. Em *8th Brazilian Symposium in Information and Human Language Technology (STIL)*, em linha.

Dyer, Chris, Victor Chahuneau & Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. Em *Conference of the North American Chapter of the Association of Computational Linguistics (NAACL)*, 644–648.

Fernandez Astudillo, Ramón, Miguel Ballesteros, Tahira Naseem, Austin Blodgett & Radu Florian. 2020. Transition-based parsing with stack-transformers. Em *Findings of the Association for Computational Linguistics: EMNLP 2020*, 1001–1007. Online: Association for Computational Linguistics.

Flanigan, Jeffrey, Sam Thomson, Jaime Carbonell, Chris Dyer & Noah A. Smith. 2014. A discriminative graph-based parser for the abstract meaning representation. Em *52nd Annual Meeting of the Association for Computational Linguistics*, 1426–1436. 10.3115/v1/P14-1134.

Garg, Sahil, Aram Galstyan, Ulf Hermjakob & Daniel Marcu. 2016. Extracting biomolecular interactions using semantic parsing of biomedical text. Em *30th AAAI Conference on Artificial Intelligence*, 2718—-2726. 10.1609/aaai.v30i1.10337.

Goodman, Michael W. 2019. AMR normalization for fairer evaluation. Em *33rd Pacific Asia Conference on Language, Information, and Computation (PACLIC)*, 37–46.

Hovy, Eduard & Julia Lavid. 2010. Towards a 'science' of corpus annotation : A new methodological challenge for corpus linguistics. *International Journal of Translation Studies* 22(1). 13–36.

Inácio, Marcio L. 2021. *Sumarização de opinião com base em abstract meaning representation*: Universidade de São Paulo — ICMC. Tese de Mestrado. 10.11606/D.55.2021.tde-13092021-141741.

Jurafsky, Dan & James H. Martin. 2009. *Speech and language processing: An introduction to natural language processing, computational linguistics and speech recognition*. Prentice Hall.

Lehmann, Fritz. 1992. *Semantic networks in artificial intelligence*. Elsevier Science Inc.

Liao, Kexin, Logan Lebanoff & Fei Liu. 2018. Abstract meaning representation for multi-document summarization. Em *27th International Conference on Computational Linguistics (COLING)*, 1178–1190.

Liu, Fei, Jeffrey Flanigan, Sam Thomson, Norman Sadeh & Noah A. Smith. 2015. Toward abstractive summarization using semantic representations. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 1077–1086. 10.3115/v1/N15-1114.

Lyu, Chunchuan & Ivan Titov. 2018. AMR parsing as graph prediction with latent alignment. Em *56th Annual Meeting of the Association for Computational Linguistics*, 397–407. 10.18653/v1/P18-1037.

Matthiessen, Christian & John A Bateman. 1991. *Text generation and systemic-functional linguistics: experiences from english and japanese*. Pinter Publishers.

Mitra, Arindam & Chitta Baral. 2016. Addressing a question answering challenge by combining statistical methods with inductive rule learning and reasoning. Em *30th AAAI Conference on Artificial Intelligence*, 2779–2785. 10.1609/aaai.v30i1.10354.

van Noord, Rik & Johan Bos. 2017. Neural semantic parsing by character-based translation: Experiments with abstract meaning representations. *Computational Linguistics in the Netherlands Journal* 7. 93–108.

Och, Franz J. & Hermann Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics* 30(4). 417–449. 10.1162/0891201042544884.

Palmer, Martha, Daniel Gildea & Paul Kingsbury. 2005. The Proposition Bank: An annotated corpus of semantic roles. *Computational Linguistics* 31(1). 71–106. 10.1162/0891201053630264.

Papineni, Kishore, Salim Roukos, Todd Ward & Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of Machine Translation. Em *40th Annual Meeting on Association for Computational Linguistics*, 311–318. 10.3115/1073083.1073135.

Peng, Xiaochang, Chuan Wang, Daniel Gildea & Nianwen Xue. 2017. Addressing the data sparsity issue in neural AMR parsing. Em *15th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 366–375.

Pourdamghani, Nima, Yang Gao, Ulf Hermjakob & Kevin Knight. 2014. Aligning English strings with abstract meaning representation graphs. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 425–429. 10.3115/v1/D14-1048.

Sheth, Janaki, Young-Suk Lee, Ramón Fernandez Astudillo, Tahira Naseem, Radu Florian, Salim Roukos & Todd Ward. 2021. Bootstrapping multilingual AMR with contextual word alignments. Em *16th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, 394–404. 10.18653/v1/2021.eacl-main.30.

Sobrevilla Cabezudo, Marco Antonio & Thiago Pardo. 2019. Towards a general abstract meaning representation corpus for Brazilian Portuguese. Em *13th Linguistic Annotation Workshop*, 236–244. 10.18653/v1/W19-4028.

Song, Linfeng & Daniel Gildea. 2019. SemBleu: A robust metric for AMR parsing evaluation. Em *57th Annual Meeting of the Association for Computational Linguistics*, 4547–4552. 10.18653/v1/P19-1446.

Song, Linfeng, Daniel Gildea, Yue Zhang, Zhiguo Wang & Jinsong Su. 2019. Semantic neural machine translation using AMR. *Transactions of the Association for Computational Linguistics* 7. 19–31. 10.1162/tacl_a_00252.

Uchida, Hiroshi, Meiying Zhu & Tarcisio Della Senta. 2006. *UNL: Universal networking language*. UNDL Foundation, International Environment House.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser & Illia Polosukhin. 2017. Attention is all you need. Em *31st International Conference on Neural Information Processing Systems*, 5998–6008.

Vieira, Thiago L. & Helena M. Caseli. 2011. PorTAl: Recursos e ferramentas de tradução automática para o Português do Brasil. Em *Brazilian Symposium in Information and Human Language Technology (STIL)*, 179–183.

Vilares, David & Carlos Gómez-Rodríguez. 2018. A transition-based algorithm for unrestricted AMR parsing. Em *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, 142–149. 10.18653/v1/N18-2023.

Xue, Nianwen, Ondrej Bojar, Jan Hajic, Martha Palmer, Zdenka Uresova & Xiuhong Zhang. 2014. Not an interlingua, but close: Comparison of English AMRs to Chinese and Czech. Em *9th International Conference on Language Resources and Evaluation (LREC)*, 1765–1772.

Zhang, Sheng, Xutai Ma, Kevin Duh & Benjamin Van Durme. 2019. AMR parsing as sequence-to-graph transduction. Em *57th Annual Meeting of the Association for Computational Linguistics*, 80–94. 10.18653/v1/P19-1009.

Zhou, Junsheng, Feiyu Xu, Hans Uszkoreit, Weiguang Qu, Ran Li & Yanhui Gu. 2016. AMR parsing with an incremental joint model. Em *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 680–689. 10.18653/v1/D16-1065.

http://www.linguamatica.com/

linguaMATICA

Artigos de Investigação