

# Un modelo simple para la detección del juicio de implicación textual entre sentencias

Saúl León, Darnes Vilariño, David Pinto, Mireya Tovar y Beatriz Beltrán

Benemérita Universidad Autónoma de Puebla,  
Facultad de Ciencias de la Computación,  
Avenida San Claudio, 14 Sur, Ciudad Universitaria.  
Puebla México.

{saul.leon,darnes,dpinto,mtovar,bbeltran}@cs.buap.mx  
<http://nlp.cs.buap.mx>

**Resumen** En el presente trabajo se propone un modelo para descubrir el juicio de implicación textual entre sentencias que incluye el juicio de contradicción. Se propone la selección de un conjunto de características léxicas, sintácticas y semánticas para la conformación de un modelo de clasificación. Los datos tanto de prueba como de entrenamiento fueron ofrecidos en el marco de la conferencia internacional Semeval 2014.

**Palabras clave:** Similitud Semántica, Implicación Textual, Medidas de Similitud Textual

## 1 Introducción

Las aplicaciones automáticas para el Procesamiento del Lenguaje Natural (PLN), necesitan un mecanismo para detectar cuando dos sentencias implican exactamente lo mismo, una es negación de la otra o cuando no existe implicación entre ellas. En concreto las tareas que más se beneficiarían con este mecanismo son: Sistemas Automáticos de Preguntas y Respuestas, Recuperación de Información, Resúmenes Automáticos, Verificación de Autoría, Atribución de Autoría, entre otras.

Desde el año 2006, resolver este problema se ha convertido en un reto y prácticamente para la mayoría de las aproximaciones desarrolladas los resultados obtenidos dependen mucho de las colecciones de datos [3]. A lo largo de los últimos 3 años, se ha propuesto la competencia SemEval<sup>1</sup> buscando encontrar propuestas que obtengan resultados significativos en corpus diferentes para este tipo de problema. En particular el modelo que se pre-

---

<sup>1</sup> <http://alt.qcri.org/semeval2014>

senta como desarrollo en esta investigación fue una propuesta de solución a la tarea 1 del Semeval 2014: Evaluación de la composición de modelos semánticos de distribución en oraciones completas a través de implicación textual, que fue considerada la subtarea 2. A continuación se explica con detalle cómo se define realmente el problema que es necesario resolver:

Tomando un par de sentencias *A* y *B*, el modelo debe determinar cuando el sentido de *B* puede inferirse a partir de *A*, es decir, *A* implica *B*. Para esta tarea en particular, el modelo debe decidir entre uno de los siguientes juicios de implicación textual:

- **Entailment:** *A* implica a *B*, ambas oraciones significan lo mismo.
- **Contradiction:** *A* contradice a *B*.
- **Neutral:** *A* es neutro con respecto a *B*, cuando ambas oraciones hablan de cosas distintas.

Por ejemplo:

- Juicio de implicación textual: **Entailment**
  - Sentencia *A*: Two teams are competing in a football match.
  - Sentencia *B*: Two groups of people are playing football.
- Juicio de implicación textual: **Contradiction**
  - Sentencia *A*: The brown horse is near a red barrel at the rodeo.
  - Sentencia *B*: The brown horse is far from a red barrel at the rodeo.
- Juicio de implicación textual: **Neutral**
  - Sentencia *A*: A man in a black jacket is doing tricks on a motorbike.
  - Sentencia *B*: A person is riding the bicycle on one wheel.

El artículo está estructurado de la siguiente manera: en la sección 2 se discuten las diferentes metodologías desarrolladas para detectar el juicio de implicación textual, en la sección 3 se presentan las características seleccionadas, se discute como se ha abordado el juicio de contradicción y se presenta el modelo propuesto. La sección 4 muestra las características de las colecciones de datos, y el análisis de los resultados obtenidos. Por último en la sección 5 se presentan las conclusiones de la investigación y el trabajo a futuro.

## 2 Trabajo Relacionado

Para resolver el problema de la implicación textual, algunos autores han recurrido a modelos que utilizan fundamentalmente conteo estadístico [11]. Estos modelos explotan diversas medidas de similitud, a partir de un par de oraciones y como resultado confeccionan vectores característicos, los vectores representativos obtenidos alimentan a un algoritmo de aprendizaje supervisado y utilizan algún modelo de clasificación. Algunas de las características que emplean son:

- Métricas Básicas (Cardinalidades de conjuntos):
  - $|A|$
  - $|B|$
  - $|A - B|$
  - $|B - A|$
  - $|A \cup B|$
  - $|A \cap B|$
  - $|A / B|$
  - $|B / A|$
- Métricas de Similitud
  - Coeficiente de Jaccard
  - Coeficiente Dice
  - Coeficiente de Sobrelapamiento
  - Pesos de Coeficientes sobrelapados
  - Similitud Coseno
  - Distancia de Manhattan
  - Distancia Euclideana
  - Distancia de Edit, a nivel de palabras
  - Distancia de Jaro-Winker
- Similitud Semántica
  - Palabra a Palabra
  - Palabra a Oración
  - Oración a Oración
  - Diferencia en las oraciones
  - Coincidencias de Palabra-Etiqueta Discurso
  - No-Coincidencias de Palabra-Etiqueta Discurso
- Relaciones Gramaticales
  - Métricas Basadas en dependencias funcionales (Stanford Parser)
  - Relaciones Parciales
  - Número de Entidades en común

En general esta forma de abordar el problema no permite ofrecer sistemas automáticos que brinden una precisión superior del 60%.

En otras investigaciones se busca el desarrollo de lo que se conoce como patrones de empatamiento [12]. Un patrón de empatamiento se basa en la aparición de términos iguales o similares que comparten un par de oraciones, por ejemplo: *Leonardo Da Vinci painted the Mona Lisa*, y *Mona Lisa is the work of Leonardo da Vinci*, donde el patrón de empatamiento es  $X \text{ paint } Y \rightarrow Y \text{ is the work of } X$ . El uso de plantillas garantiza que la implicación textual será descubierta, no obstante estos patrones de empatamiento son difíciles de construir, ya que son muy particulares. En la actualidad se ha propuesto una

metodología no supervisada para generar patrones de empatamiento sobre un large-scale-corpus [7].

Una vertiente nueva intenta generalizar los patrones extraídos logic-based [1], a continuación se muestra un ejemplo:

- $S_1$ : Leonardo Da Vinci painted the Mona Lisa
- $S_2$ : Mona Lisa is the work of Leonardo da Vinci
- Equivalencia de  $S_1$ :  $isPainterOf(DaVinci, MonaLisa)$
- Equivalencia de  $S_2$ :  $isWorkOf(MonaLisa, DaVinci)$
- Regla generada:  $\forall x \forall y (isPainterOf(x, y) \Rightarrow isWorkOf(y, x))$

Estos axiomas introducidos, hasta el momento aún no logran detectar la idea que transmiten las sentencias, este problema también ha sido abordado utilizando técnicas de preguntas y respuestas, considerando que  $S_1$  es la pregunta y  $S_2$  es la respuesta. Esta metodología también presupone de alguna manera que en la respuesta aparecen términos de la pregunta [6].

En los últimos años se han desarrollado recursos léxicos que han permitido hacer un estudio semántico entre las sentencias, por ejemplo en el trabajo desarrollado por [4] se propone un sistema de reconocimiento de inferencias que utilizan herramientas como: PoS taggers y Dependency Parser, y recursos léxicos tales como: WordNet y Thesaurus de sinónimos.

El problema de todas estas aproximaciones radica en que se extraen características de los datos de entrenamiento, que no necesariamente aparecen en los datos de prueba, lo que dificulta mucho detectar si existe o no implicación textual entre las sentencias. Por otra parte, desde hace varios años, se han introducido la representación mediante grafos para diversas tareas del procesamiento del lenguaje natural, ya que por su estructura permite relacionar de manera correcta y en diferentes niveles la relación entre los términos que forman un determinado texto [5, 10], pero a pesar de los esfuerzos realizados los resultados obtenidos continúan dependiendo del corpus de trabajo.

Todas las metodologías para detectar el juicio de implicación textual han trabajado fundamentalmente sobre los juicios *Entailment* y *No-Entailment*, sin embargo por primera vez se brinda un corpus para detectar el juicio de implicación textual *Contradiction*. En esta investigación se propone un conjunto de características a extraer del par de sentencias y se desarrolla un algoritmo que permite fácilmente detectar el juicio de implicación textual *Contradiction*. La metodología se explica en la próxima sección.

### 3 Metodología propuesta

La metodología propuesta para la detección del juicio de implicación se ha dividido en 2 partes. En la primera parte se propone un conjunto de características que se extraen de ambas oraciones, intentando con esto detectar el sentido de cada una de ellas. En la se-

gunda parte se propone la inclusión de dos características que permiten detectar a través de un algoritmo el juicio de implicación *Contradiction*. Las características extraídas se realizan a nivel de palabras y de caracteres, además se proponen medidas de similitud semántica basadas en conocimiento y medidas de similitud semántica basadas en corpus (Ver tabla 1).

### 3.1 Extracción de Características

A partir de cada par de oraciones, se extraen los  $n$ -gramas de caracteres de 1 a 5 que comparten; así como los *skip*-grama que comparten. Los *skip*-gramas extraídos son saltos a nivel de caracteres sobre cada palabra con intervalos de 1 a 5. También se cuantifican los  $n$ -gramas y *skip*-gramas de las palabras que comparten, con longitud de 1 a 5. La medida de similitud de Jaccard expandida, Un modelo simple para la detección del juicio de implicación textual 5 intenta no solamente detectar si se comparten palabras en ambas oraciones, sino detectar también si ambas oraciones comparten sinónimos [2]. Utilizando la herramienta NLTK2 se han podido introducir las 6 medidas de similitud semántica basadas en conocimiento que esta ofrece. Se incluye además, la métrica ofrecida por Rada [9] en la que la similitud entre cada par de palabras de una oración con respecto a la otra es por medio de PMI y LSA. Esto hace un total de 29 características.

Tabla 1. Características extraídas

Característica	Tipo
$n$ -gramas de caracteres y palabras	Léxica
<i>skip</i> -gramas de caracteres y palabras	Léxica
Coficiente de Similitud de Jaccard con sinónimos	Léxica
Similitud de Leacock & Chodorow a nivel de palabras	Basada en conocimiento
Similitud de Lesk a nivel de palabras	Basada en conocimiento
Similitud de Wu & Palmer a nivel de palabras	Basada en conocimiento
Similitud de Resnink a nivel de palabras	Basada en conocimiento
Similitud de Lin a nivel de palabras	Basada en conocimiento
Similitud de Jiang & Conrath a nivel de palabras	Basada en conocimiento
Métrica de Rada Mihalcea usando PMI	Basada en corpus
Métrica de Rada Mihalcea usando LSA	Basada en corpus

Con las características anteriormente expuestas no es tan simple detectar cuando el juicio de implicación textual es *Entailment* o *Contradiction*, es por ello que se propone agregar otras 2 nuevas características, que se discuten a continuación.

<sup>2</sup> <http://www.nltk.org>

### 3.2 Tratamiento de la Implicación Textual

Se busca la aparición de palabras que denotan contradicción en alguna de las 2 oraciones. Se comienza analizado si en alguna de las oraciones existe alguna palabra que tenga como sufijo: *n't*, además se busca la aparición de algunas de las siguientes palabras: no, not, none y nobody. Además como pueden existir pares de oraciones que se nieguen totalmente sin considerar la negación de manera explícita, se propone introducir la detección de la presencia de antónimos en alguna de ellas. Esto se resuelve de la siguiente manera a una variable se le asigna el valor 1 si en alguna de las oraciones que conforman el par aparece alguna de las 4 palabras o algún antónimo en la oración complementaria. Si no hay aparición alguna de estos elementos, a la variable se le asigna el valor 0.

Para reforzar las características extraídas anteriormente, para el caso de juicio *Entailment*, se le asigna a una variable 1 si en alguna de las 2 oraciones hay presencia de sinónimos, en caso contrario dicha variable almacena el valor 0. La descripción de este proceso se muestra en el Programa 1.

---

**Programa 1** Detección de implicación textual.

---

*Detectar Empatamiento*

```
program detectar_empatamiento(esNegado: Integer; esSinonimo: Integer)
var
  S_1: Sentencia 1
  S_2: Sentencia 2
  sinonimos: Diccionario de Sinonimos
  antonimos: Diccionario de Antonimos
  negacion: Diccionario con las palabras: no, not, none y nobody
begin
  esNegado := 0;
  esSinonimo := 0;
  H_1 := Conjunto(Lematizar(S_1));
  H_2 := Conjunto(Lematizar(S_2));
  H_interseccion := H_1.interseccion(H_2);
  H_1 := H_1 - H_interseccion
  H_2 := H_2 - H_interseccion

  for T_i in H_1
    if T_i in negacion OR TerminaCon(T_i, "n't")
      esNegado = 1
  for T_i in H_2
    if T_i in negacion or TerminaCon(T_i, "n't")
```

```
esNegado = 1

if esNegado == 0
  for T_i in H_1
    for T_j in H_2
      if T_i in antonimos[T_j] OR T_j in antonimos[T_i]
        esNegado = 1;
        break;
  for T_i in H_1
    for T_j in H_2
      if T_i in sinonimos[T_j] or T_j in sinonimos[T_i]
        esSinonimo = 1;
        break;

return esNegado, esSinonimo;
end.
```

---

Finalmente se confecciona un vector representativo de cada par de sentencias con 31 características, y se propone desarrollar un modelo de clasificación, que considere cada una de las clases que conforman los juicios de implicación textual dados en el corpus de entrenamiento. El modelo se desarrolla utilizando como clasificador la máquina de soporte vectorial. Una vez construido el modelo se clasifican los datos de prueba.

## 4 Resultados experimentales

A continuación se describen los datos que permitieron validar el modelo desarrollado.

### 4.1 Conjunto de Datos

Se parte de dos conjuntos de datos, uno de entrenamiento y otro de prueba. Los datos fueron proporcionados como parte de la tarea 1, subtarea 2, del SemEval-2014 [8]. Los datos de entrenamiento están compuestos por: 2,536 pares de oraciones neutrales, 1,299 pares con empatamiento y 665 oraciones contradictorias.

Para probar el modelo desarrollado se ofreció un corpus formado por 4,927 pares de oraciones.

### 4.2 Resultados Obtenidos

Esta propuesta fue enviada a la conferencia SemEval-2014 y en la tabla 2 se muestran los resultados alcanzados. Nuestra propuesta se reconoce como **BUAP\_run1**.

El modelo desarrollado obtuvo el séptimo lugar de 18 equipos participantes, con una precisión de más del 79%. A pesar de que no se sabe cuántas categorías reales se han detectado se puede decir que el desempeño de la propuesta es bueno. En el proceso de validación de la propuesta desarrollada, aplicamos para entrenar validación cruzada a 10 pliegues, y se detectó que la inclusión de las últimas 2 características (esNegado y esSinonimo) permitió detectar con una precisión del 92% el juicio de *Contradiction* y con una precisión del 80% al juicio de *Entailment*.

## 5 Conclusiones y recomendaciones

En el presente trabajo se describe la propuesta enviada por nuestro grupo de investigación a la conferencia internacional SemEval-2014, tarea 1 subtarea 2.

El desempeño logrado ha sido muy satisfactorio, si se piensa que solamente se ha trabajado con vectores representativos de cada par de oraciones que incluyen 31 características. Se ha experimentado con características de diferentes tipos, a nivel de caracteres, a nivel de palabras, métricas basadas en conocimiento y métricas basadas en corpus. Los resultados obtenidos permiten determinar que las características extraídas le han permitido al sistema obtener resultados importantes, prácticamente con un 80% de precisión.

**Tabla 2.** Resultados de la Tarea 1, subtarea 2 del SemEval-2014

ID del Equipo	Precisión
Illinois-LH_run1	84.575
ECNU_run1	83.641
UNAL-NLP_run1	83.053
SemantiKLUE_run1	82.322
The_Meaning_Factory_run1	81.591
CECL_ALL_run1	79.988
<b>BUAP_run1</b>	<b>79.663</b>
UoW_run1	78.526
CDT_run1	77.106
UIO-Lien_run1	77.004
FBK-TR_run3	75.401
StanfordNLP_run5	74.488
UTexas_run1	73.229
Yamraj_run1	70.753
asjai_run5	69.758
haLF_run2	69.413
CNGL_run1	67.201
UANLPCourse_run2	48.731

Sin embargo, aún no se han introducido mecanismos de inferencia, que de alguna manera le permita al sistema detectar que hay pares de oraciones que comparten mucho y no existe implicación textual entre ellas y hay pares de oraciones que no comparten nada y expresan exactamente lo mismo. Como trabajo a futuro se considera incluir vectores de contexto para estudiar el grado de correlación entre las diferentes palabras que conforman todo el corpus, y patrones de empatamiento (*matching patterns*) entre los pares de sentencias.

## **Referencias**

1. Johan Bos. Is there a place for logic in recognizing textual entailment? *Linguistic Issues in Language Technology*, 9, (2013)
2. Maya Carrillo, Darnes Vilarino, David Pinto, Mireya Tovar, Saul León, and Esteban Castillo. Buap: three approaches for semantic textual similarity. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, pages 631–634. Association for Computational Linguistics, (2012)
3. Ido Dagan, Oren Glickman, and Bernardo Magnini. The pascal recognising textual entailment challenge. In *Machine learning challenges. evaluating predictive uncertainty, visual object classification, and recognising textual entailment*, pages 177–190. Springer, (2006)
4. Min-Yuh Day, Chun Tu, Shih-Jhen Huang, Hou Cheng Vong, and Sih-Wei Wu. Imtku textual entailment system for recognizing inference in text at ntcir-10 rite-2. In *Proceedings of the 10th NTCIR Conference*, (2013)
5. Aria D Haghighi, Andrew Y Ng, and Christopher D Manning. Robust textual inference via graph matching. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 387–394. Association for Computational Linguistics, (2005)
6. Sanda Harabagiu and Andrew Hickl. Methods for using textual entailment in open-domain question answering. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 905–912. Association for Computational Linguistics, (2006)
7. Milen Kouylekov and Bernardo Magnini. Building a large-scale repository of textual entailment rules. In *Proceedings of LREC*, (2006)
8. Marco Marelli, Stefano Menini, Marco Baroni, Luisa Bentivogli, Raffaella Bernardi, and Roberto Zamparelli. A sick cure for the evaluation of compositional distributional semantic models. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry

- Declerck, Hrafn Loftsson, Bente Maegaard, Joseph Mariani, Asuncion Moreno, Jan Odijk, and Stelios Piperidis, editors, Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), Reykjavik Iceland, European Language Resources Association (ELRA). may (2014)
9. Rada Mihalcea, Courtney Corley, and Carlo Strapparava. Corpus-based and knowledge-based measures of text semantic similarity. In *AAAI*, volume 6, pages 775–780, (2006)
  10. Rada Mihalcea and Dragomir Radev. *Graph-based natural language processing and information retrieval*. Cambridge University Press, (2011)
  11. Jiang Zhao, Man Lan, and Zheng-yu Niu. Ecnucs: Recognizing cross-lingual textual entailment using multiple text similarity and text difference measures. Atlanta, Georgia, USA, page 118, (2013)
  12. Shiqi Zhao, Xiang Lan, Ting Liu, and Sheng Li. Application driven statistical paraphrase generation. In Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL Volume 2, pages 834–842. Association for Computational Linguistics, (2009)