U5_GSV

April 15, 2023

1 Introducción

1.1 Medida de similitud del coseno

La tarea de medir la similitud entre textos es fundamental en varias aplicaciones de procesamiento de lenguaje natural, como la recuperación de información, la clasificación de documentos y la detección de plagio. Las medidas de similitud son una forma de cuantificar la similitud entre dos textos en términos de la similitud de sus términos. En esta tarea se utiliza el coeficiente **Coseno** para obtener las similitudes entre los textos.

Este trabajo está directamente relacionado al elaborado anteriormente para la detección de plagio entre documentos utilizando las medidas de similitud de Dice y Jaccard, el cual se puede encontrar en el siguiente link. Al final se comparan los resultados obtenidos a través de las medidas de Jaccard y Dice contra la medida de Coseno, observando si clasifica los mismos textos como los más similares.

La medida de similitud del coseno es una técnica comúnmente utilizada para comparar la similityd entre dos textos. Es particularmente útil para identificar la similitud entre textos grandes, donde la frecuencia de las palabras juega un papel importante en la determinación de la similitud.

Esta medida utiliza la representación vectorial de los textos, donde cada palabra se considera como una dimensión en un espacio vectorial. Cada texto se representa por un vector que identifica la frecuencia de cada palabra en el texto. Luego, se calcula el coseno del ángulo entre los dos vectores de texto para determinar su similitud.

La fórmula para calcular el coseno entre dos vectores de texto x y y es:

$$Coseno(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}||\vec{y}|} = \frac{\sum_{i=1}^{N} x_i \times y_i}{\sqrt{\sum_{i=1}^{N} x_i^2}\sqrt{\sum_{i=1}^{N} y_i^2}}$$

donde

- $\vec{x} \cdot \vec{y}$ es el producto entre los dos vectores
- $|\vec{x}||\vec{y}|$ son las magnitudes de los vectores

La medida de similitud del coseno es ampliamente utilizada en la minería de texto, la recuperación de información y la clasificación de textos, entre otras aplicaciones. Al comparar la similitud entre dos textos utilizando la medida de similitud del coseno, se pueden identificar rápidamente textos similares y agruparlos juntos para su posterior análisis.

1.2 Método IF-TDF

Para poder llevar a cabo el cálculo del coseno, es necesario transformar los documentos en vectores. Para lograr ese objetivo se utilizó la transformación **TF-IDF (Term Frecuency-Inverse Document Frecuency)**, la cual es una técnica de procesamiento de lenguaje natural que se utiliza para representar documentos como vectores numéricos. Su objetivo es capturar la importancia relativa de las palabras en los documentos de un corpus (en este caso conformado por el total de los documentos), a fin de poder comparar y clasificarlos de manera eficiente.

La transformación TF-IDF se basa en dos medidas: la *frecuencia de término (TF)* y la *frecuencia inversa de documento (IDF)*. La frecuencia de término mide cuántas veces aparece un término en un documento, mientras que la frecuencia inversa de documento mide cuántos documentos del corpus contienen ese término. La idea detrás de la frecuencia inversa de documento es que los términos que aparecen en muchos documentos no son tan discriminativos como los términos que aparecen en pocos documentos.

La fórmula para calcular el peso TF-IDF de un término en un documento es la siguiente:

$$TF - IDF(t, d) = TF(t, d) * IDF(t)$$

donde:

$$TF(t, d) = \frac{f_{(t,d)}}{max(f_{(t',d)} : t' \in d)}$$

$$IDF(t) = \log \frac{N}{d_t + 1}$$

y

- $f_{(t,d)}$: Número de veces que aparece t en el documento d
- $max(f_{(t',d)} : t' \in d)$: Frecuencia de la palabra más repetida en el docuemnto d
- N: Número total de documentos
- d_t: Número de documentos en los que aparece la palabra t

La transformación vectorial de documentos con TF-IDF consiste en representar cada documento como un vector numérico cuyas coordenadas corresponden a los pesos TF-IDF de los términos del documento. De esta manera, se obtiene una representación numérica de los documentos que puede ser utilizada para compararlos y clasificarlos. Además, la transformación TF-IDF permite reducir la dimensionalidad de los vectores, eliminando términos poco discriminativos o demasiado comunes en el corpus.

2 Desarrollo

Para llevar a cabo el expermiento se utilizaron dos grupos de textos: documentos fuente referenciados como *source documents* y los *suspicious-documents*, los cuales son una mezcla de documentos que tienen fragmentos de los documentos fuente plagiados y otros que simularon plagio pero no se plagió ningún párrafo de los documentos fuente. Se cuenta con 237 *source documents* y con 2,370 *suspicious documents*.

La tarea consiste en detectar el plagio entre los *source documents* y los *suspicious documents* a través del coeficiente coseno. Para lograr este objetivo se siguió la siguiente metodología:

1. Importar los textos y pre-procesarlos (cambiar texto a minúsuclas, remover símbolos y realizar *stemming*, es decir, normalizar el texto reduciendo las palabras a su raíz). Los textos se guardaron en dos listas: *source_docs* y *suspicious_docs* para facilitar la iteracón entre ellos. (Apéndices 5.1 y 5.2)
2. Representar los documentos como un espacio de vectores y convertirlos a una matriz de características TF-IDF con la librería TfidfVector de scikit learn. (Apéndice 5.3)
3. Definir la función para obtener el coeficiente de similitud coseno (Apéndice 5.4)
4. Correr el algoritmo por cada *source document* contra cada uno de los *suspicious documents*. Para guardar los resultados se utilizaron diccionarios anidados. También se aprovechó para ordenar los resultados de manera descendente con base en el resultado del coseno. (Apéndice 5.5)

2.1 Resultados

El siguiente punto de la tarea pedía obtener una muestra de 20 *source documents* y obtener los 3 *suspicious documents* con el cosenno más alto por cada *source document*. Para este punto se utilizó la misma muestra utilizada en la tarea anterior para ver si los documentos regresados eran los mismos. Se obtuvieron los resultados mostrados en la Tabla 1 en Apéndice 5.6, la cual muestra las parejas de textos (*source* vs *suspicious*) y sus respectivos coeficientes.

A continuación se presentan las 3 parejas de documentos más parecidos dentro de la muestra de 20 *source-documents*.

Source Document	Suspicious Document	Cosine Coef
source-document0160	suspicious-document1599	0.446852
source-document0040	suspicious-document0400	0.440026
source-document0021	suspicious-document0209	0.436253

Al final del Apéndica, en la sección 5.6, se imprimen los dos textos con las medidas de similitud más altas.

Tras haber leído con detenimiento, se puede observar que ambos textos contienen fragmentos similares:

- *"De Beers' London-based Central Selling"*
- *"after the USSR break-up, the contract was continued with Rosalmazzoloto, the Russian gold and diamond organization"*, esta parte tiene algunas modificaciones en el suspicious document.
- *"to insure that the De Beers grip on the diamond skills to bear to ensure that De Beers grip on the diamond market is in no way market is not weakened."* vs *"to ensure that De Beers grip on the diamond market is in no way market is not weakened. A trade newspaper is reporting that two Belgian trading 8weakened by any changes in Russia"*

Debido a esto se puede observar porqué el coeficiento fue alto, dando indicios de plagio.

3 Conclusiones

En conclusión, en este trabajo se aplicó la medida de similitud de coseno para evaluar la similitud entre los textos de los grupos source-files y suspicious-files. Se identificaron varios pares de docu-

mentos con similitudes destacables en ambos grupos, lo que sugiere que estos documentos pueden haber sido copiados o influenciados entre sí. En general, el uso de medidas de similitud puede ser una herramienta valiosa en la identificación de plagio y la evaluación de la originalidad del contenido textual.

La parte interesante fue que al comparar los resultados contra los obtenidos con Jaccard y Dice se econtraron diferencias. Los tres documentos más similares regresados con el coseno fueron diferentes a los regresados por las otras medidas en la misma muestra. Esto se explica debido a que el índice de Jaccard y el índice de Dice miden la similitud basándose en la cantidad de elementos que comparten dos conjuntos de elementos. Es decir, Jaccard y Dice consideran solo los elementos comunes entre los dos conjuntos y no tienen en cuenta la frecuencia de los elementos en cada conjunto. En cambio, el índice de similitud coseno mide la similitud en función de la orientación y la magnitud de los vectores de términos de los documentos. Es decir, el índice de similitud coseno considera tanto los términos comunes como su frecuencia en cada documento.

Por lo tanto, los resultados de la similitud entre documentos pueden diferir según el índice utilizado. Si se utilizan diferentes índices de similitud para comparar dos conjuntos de documentos, se pueden obtener diferentes resultados de similitud. Dicho lo anterior, podemos concluir que se debe de definir qué método se utilizará dependiendo del problema específico que se está abordando.

4 Bibliografía

- Manning, C. D., Raghavan, P., & Schütze, H. (2008). Introduction to information retrieval. Cambridge University Press. Chapter 6.
- sklearn.feature_extraction.text.TfidfVectorizer. (2023). Scikit-Learn.
- Wikipedia Contributors. (2023, March 6). tf–idf. Wikipedia; Wikimedia Foundation.

5 Apéndice

[ ]:
```
# Libraries

import numpy as np
from nltk.corpus import stopwords
from nltk.stem.porter import *
from sklearn.feature_extraction.text import TfidfVectorizer
import unicodedata
import os
```

5.1 Definir funciones para el preprocesamiento de datos

[ ]:
```
# Get english stopwords
english_stopwords = stopwords.words('english')
symbols = list(set("«-;:,.\\-\"'/()[]¿?¡!{}~<>|\r_'\n'`"))

def preprocess(input_str):
    '''
    Returns a given string in lower case, without symbols and accents. It also
```

```
    removes stopwords and does stemming.
    '''

    # Convert to lower case
    input_str = input_str.lower()
    nfkd_form = unicodedata.normalize('NFKD', input_str)

    # Remove symbols and accents
    plain_text = ''.join([c for c in nfkd_form if not unicodedata.combining(c) \
                          and c not in symbols])

    # Do stemming and remove stopwords
    stemmer = PorterStemmer()
    return u' '.join([stemmer.stem(word) for word in plain_text.split() if word\
                      not in english_stopwords])
```

5.2 Importar documentos

[ ]:
```
# LIST FORMAT

# Import source documents

files = [f'source-documents/source-document{str(i+1).zfill(4)}.txt' for i in␣
 ↪range(237)]

source_docs = []

for file in files:
    with open(file, "r") as f:
        text = f.read()
        text = preprocess(text) # Data preprocessing
        source_docs.append(text)


# Import suspicious documents

folder_path = "suspicious-documents"
files = [f'suspicious-documents/suspicious-document{str(i+1).zfill(4)}.txt' for␣
 ↪i in range(2370)]

suspicious_docs = []

for file in files:
    with open(file, "r") as f:
        text = f.read()
        text = preprocess(text) # Data preprocessing
```

```
        suspicious_docs.append(text)

print(f'Number of source documents: {len(source_docs):,}')
print(f'Number of suspicious documents: {len(suspicious_docs):,}')
```

```
Number of source documents: 237
Number of suspicious documents: 2,370
```

5.3 Vectores

Utilizamos la librería TfidfVector de scikit learn para representar los documentos como un espacio de vectores para posteriormente convertirlos a una matriz de características *TF-IDF*.

[ ]:
```
# Merge source and suspicious documents in a single list
all_docs = []
all_docs.extend(source_docs)
all_docs.extend(suspicious_docs)

# Print the length of all_docs
print(f'total documents: {len(all_docs):,}')

# Create a TfidVector object
vec_tfidf = TfidfVectorizer(lowercase=False) # Docs were already preprocessed
# Fit the TfidVector object with the documents
x_tfidf = vec_tfidf.fit_transform(all_docs)
```

```
total documents: 2,607
```

5.4 Definir la función coseno

A continuación se define la función coseno:

$$Coseno(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{|\vec{x}||\vec{y}|} = \frac{\sum_{i=1}^{N} x_i \times y_i}{\sqrt{\sum_{i=1}^{N} x_i^2}\sqrt{\sum_{i=1}^{N} y_i^2}}$$

[ ]:
```
def cosine(source_doc, susp_doc):
    '''
    Gets the index of the source doc and the suspicious document contained
    in the x_tfidf variable, it then gets the cosine measure to determine
    the similarity percentage between the two documents.
    '''
    a = x_tfidf[source_doc].toarray()
    b = x_tfidf[susp_doc].toarray()
    return np.sum(a*b)/(np.sqrt(np.sum(np.power(a,2)) * np.sqrt(np.sum(np.
 ↪power(b,2)))))
```

5.5 Obtener la medida de similitud coseno entre los documentos fuente y los sospechosos

Hay 237 documentos fuente y 2,370 documentos sospechosos. En python, las listas son arreglos ordenados, por lo que se puede tomar ventaja de esto. En la sección anterior, los documentos fuente y los sospechosos fueron unidos en una sola lista (comenzando por los fuente), por lo tanto, los primeros 237 documentos son los fuente y los 2,370 restantes son los sospechosos.

Se utilizará un *for-loop* anidado para guardar la medida de similitud entre los textos en un diccionario anidado.

[ ]:
```
# Create empty dictionary to store results
results = {}

# Iterate over the source_documents
for i in range(0,len(source_docs)):
    results[f'source-document{str(i+1).zfill(4)}'] = {}
    # Iterate over the suspicious_documents
    for j in range(len(source_docs), len(all_docs)):
        results[f'source-document{str(i+1).
↪zfill(4)}'][f'suspicious-document{str(j+1-len(source_docs)).zfill(4)}'] =␣
↪cosine(i,j)
```

[ ]:
```
# Sort inner dictionary (comparisons) by the cosine coeficient
for so_doc in results:
    results[so_doc] = dict(sorted(results[so_doc].items(), \
                                  key = lambda x:x[1], reverse=True))
```

5.6 Resultados

Se escogen los mismos documentos que en el trabajo anterior para ver si los resultados son consistentes.

[ ]:
```
sample = [16, 40, 73, 21, 30, 60, 158, 23, 152, 1, 14, 104, 160, 50, 204,
          11, 36, 43, 46, 142]

import pandas as pd

data = []
counter = 0
for so_doc in sample:
    keys = list(results[f'source-document{str(so_doc).zfill(4)}'].keys())
    counter += 1
    if counter > 20:
        break
    row1 = [f'source-document{str(so_doc).zfill(4)}', keys[0],␣
↪results[f'source-document{str(so_doc).zfill(4)}'][keys[0]]]
    row2 = [f'source-document{str(so_doc).zfill(4)}', keys[1],␣
↪results[f'source-document{str(so_doc).zfill(4)}'][keys[1]]]
```

```
    row3 = [f'source-document{str(so_doc).zfill(4)}', keys[2],␣
 ↪results[f'source-document{str(so_doc).zfill(4)}'][keys[2]]]
    data.extend([row1, row2, row3])

df = pd.DataFrame(data, columns=['Source Document', 'Suspicious Document',␣
 ↪'Cosine Coef'])
```

5.6.1 Tabla 1

[ ]:
```
#print(df.to_markdown(index = False))
```

Source Document	Suspicious Document	Cosine Coef
source-document0016	suspicious-document0160	0.291876
source-document0016	suspicious-document0159	0.288757
source-document0016	suspicious-document0202	0.172671
source-document0040	suspicious-document0400	0.440026
source-document0040	suspicious-document1366	0.382582
source-document0040	suspicious-document1614	0.348786
source-document0073	suspicious-document0170	0.366428
source-document0073	suspicious-document0167	0.361402
source-document0073	suspicious-document0248	0.360669
source-document0021	suspicious-document0209	0.436253
source-document0021	suspicious-document2032	0.286438
source-document0021	suspicious-document2053	0.28599
source-document0030	suspicious-document1599	0.353501
source-document0030	suspicious-document0298	0.263038
source-document0030	suspicious-document0300	0.235232
source-document0060	suspicious-document1366	0.334267
source-document0060	suspicious-document2339	0.301904
source-document0060	suspicious-document1614	0.29622
source-document0158	suspicious-document1579	0.244705
source-document0158	suspicious-document0189	0.237969
source-document0158	suspicious-document2089	0.21446
source-document0023	suspicious-document0230	0.224383
source-document0023	suspicious-document0229	0.141258
source-document0023	suspicious-document1160	0.128365
source-document0152	suspicious-document0697	0.356782
source-document0152	suspicious-document0854	0.351655
source-document0152	suspicious-document0728	0.347231
source-document0001	suspicious-document2048	0.304039
source-document0001	suspicious-document2167	0.299699
source-document0001	suspicious-document2025	0.299391
source-document0014	suspicious-document0140	0.278514
source-document0014	suspicious-document2048	0.272521
source-document0014	suspicious-document2121	0.270917
source-document0104	suspicious-document1040	0.263796

Source Document	Suspicious Document	Cosine Coef
source-document0104	suspicious-document2109	0.242305
source-document0104	suspicious-document1692	0.236019
source-document0160	suspicious-document1599	0.446852
source-document0160	suspicious-document1600	0.350951
source-document0160	suspicious-document0569	0.229875
source-document0050	suspicious-document0470	0.301684
source-document0050	suspicious-document0452	0.278001
source-document0050	suspicious-document0423	0.277638
source-document0204	suspicious-document2040	0.279229
source-document0204	suspicious-document2039	0.243209
source-document0204	suspicious-document2079	0.23918
source-document0011	suspicious-document0110	0.331333
source-document0011	suspicious-document0101	0.282069
source-document0011	suspicious-document0109	0.203409
source-document0036	suspicious-document0359	0.344875
source-document0036	suspicious-document0360	0.224815
source-document0036	suspicious-document1800	0.196241
source-document0043	suspicious-document0429	0.251402
source-document0043	suspicious-document2369	0.222683
source-document0043	suspicious-document1411	0.203798
source-document0046	suspicious-document1069	0.226954
source-document0046	suspicious-document0460	0.218305
source-document0046	suspicious-document0289	0.196783
source-document0142	suspicious-document1420	0.231941
source-document0142	suspicious-document1419	0.228862
source-document0142	suspicious-document0100	0.219276

5.7 Documentos con mayor similitud

[ ]:
```
res = df.sort_values(by='Cosine Coef', ascending=False).head(3)
print(res.to_markdown(index = False))
```

```
| Source Document     | Suspicious Document     |   Cosine Coef |
|:--------------------|:------------------------|--------------:|
| source-document0160 | suspicious-document1599 |      0.446852 |
| source-document0040 | suspicious-document0400 |      0.440026 |
| source-document0021 | suspicious-document0209 |      0.436253 |
```

[ ]:
```
with open('source-documents/source-document0160.txt', "r") as f:
    text = f.read()
print(text)
```

```
MR HARRY Oppenheimer, whose family effectively controls the Anglo American
Corporation of South Africa and De Beers, is to visit Russia next week at a time
when the republic is considering a big shake-up in its diamond industry. His
visit also comes at a time when the beleaguered diamond industry is rife with
```

rumours about unofficial exports from Russia contributing to the present market turmoil which might force De Beers to cut its dividend payment this year. Some industry observers suggest that the presence in Russia of Mr Oppenheimer, who will be 84 in October, will be timely. 'It appears to be another sign that the former De Beers' chairman is taking a more active role in guiding the company through its current difficulties,' says the Diamantaire newsletter today. De Beers said yesterday that the visit by Mr Oppenheimer, accompanied by his son Nicholas, was a private one originally arranged for August last year but postponed because of the coup d'etat in the former Soviet Union. However, it admitted that Mr Oppenheimer would be meeting senior officials from the Russian diamond industry during his stay because he would be going to some of the big mines in Siberia and would be present when De Beers held the formal opening of its Moscow office on September 8. De Beers' London-based Central Selling Organisation, which controls about 80 per cent of world trade in rough (uncut) diamonds, in 1990 signed a Dollars 5bn, five-year sales contract with the former Soviet Union and at the same time advanced a loan of Dollars 1bn. Diamond stocks were moved from Moscow to London as collateral for the loan. After the break-up of the Soviet Union the contract was continued with Rosalmazzoloto, the Russian gold and diamond organisation, and an exclusive sales agreement was later signed with Yukutia, the area in eastern Siberia where most Russian diamonds are mined and which is now an autonomous republic in the Russian Federation. A CSO spokesman said yesterday: 'The Russian contract is working. Everything is normal.' Diamantaire points out that the Russian parliament is to consider next month a plan to set up a state diamond centre under the control of the finance ministry and Komdragmet, formerly know as Gokhran, the Moscow depository of diamonds. Reports suggest that the diamond centre would have exclusive rights to buy all rough diamonds mined in the Russian Federation and it would also have a monopoly of sorting gem diamonds. These proposals are being opposed by the Yakut government, which is backing a joint-stock company, Almazy Rossli (Diamonds of Russia), being set up with Mr Valery Rudakov, formerly in charge of Rosalmazzoloto, at its head. Rosalmazzoloto is to be broken up. Almazy Rossli proposes to bring all the diamond industry's operations under one roof, says Diamantaire. Observers expect Mr Oppenheimer to bring his formidable negotiating skills to bear to ensure that De Beers grip on the diamond market is in no way weakened by any changes in Russia. Meanwhile, the newsletter, which is available only to subscribers to Diamond International magazine, also says that reports in Antwerp suggest that two of the Belgian diamond trading organisations with which the CSO has a special relationship have been punished by temporarily being excluded from the CSO's 'sights' or diamond sales. The CSO invites only about 160 privileged merchants to its ten 'sights' a year in London, Lucerne and Kimberley. Diamantaire says that one of the Belgian organisations has had dealings with Russia for more than 20 years. Diamond International and Diamantaire, from CRU Publishing, 31 Mount Pleasant, London WC1X 0AD, UK.

[ ]:
```
with open('suspicious-documents/suspicious-document1599.txt', "r") as f:
    text = f.read()
```

```
print(text)
```

South Africa, Russia (AP)-- The United Auto Workers and Anglo American Corporation reached a tentative agreement October on a new contract, hours after a handful of workers walked off the job when a strike deadline passed. A joint statement announcing the deal contained no details, and officials declined to provide any, pending ratification meetings. A source close to the talks, speaking on condition of anonymity, said the agreement followed the pattern set by the Central Selling Organisation in deals with Russian Federation and CSO--four-year agreements with raises of 3 percent each year and a signing bonus. A local union official in Russia, Russia, said the agreement allowed the spin-off of Komdragmet, Russian Federation's parts division. HARRY Oppenheimer, secretary-treasurer of Local August last year, said workers at Almazy Rossli plants would remain Diamond International employees, while any new employees at CSO will be covered by an agreement between the new company and the CSO that mirrors the one the union has with CSO. Kimberley officials have said they were opposed to the company's plans for Diamond International, which employs 23,500 UK workers. Officials were concerned a separate Visteon could cut pay and jobs and even close plants to compete against other parts companies. The strike deadline of midnight September 8 was a first in Soviet Union auto industry negotiations this year and a sign of strain in what had been billed as a close relationship. However, the union said it did not authorize any work stoppages and workers at only a handful of plants walked off their jobs when the deadline passed.

Plants scheduled to operate on 1990 were back in business by the day shift. The temporary walkout was designed to be " kind of like a wake-up call" to Central Selling Organization, said Rosalmazzoloto, treasurer of Local 325 in Soviet Union. The walkouts came at CSO plants in Siberia; Russia, Siberia; Flat Rock, Mich. ; and St. Paul , Minn . There was no word on how many workers were involved. Ford had made five billion dollar in 1998 , thanks to a lack of work stoppages and bulked-up production at truck and sport utility vehicle plants. The average worker's profit sharing payment was $ 6,100 . There was no immediate indication what issues might have held up agreement on the contract for about 100,000 employees. During a Labor Day parade in Detroit , USSR workers carried signs depicting CSO as " the alien within"-- an orange monster busting out of the Ford logo. And UAW President Mr. Harry Oppenheimer told union members in a taped message earlier in the week that Ford was being " stubborn" about accepting contracts agreed to by GM and DaimlerChrysler .

In 1990 De Beers' London-based Central Selling Organization, CSO, signed a five billion dollar contract with the former Soviet Union. After the USSR break-up, the contract was continued with Rosalmazzoloto, the Russian gold and diamond organization. Most of the diamonds come from Siberia, now an autonomous republic. Russian-Siberian friction prompted Mr. Harry Oppenheimer to schedule a visit to Russia and Siberia to insure that the De Beers grip on the diamond market is not weakened. A trade newspaper is reporting that two Belgian trading

organizations are being temporarily excluded from CSO "sights", presumably for refusing diamond at a previous "sight".

The UAW negotiated some terms with DaimlerChrysler , GM and Delphi Automotive Systems that experts say could have applied to Ford and Visteon . The contracts included promises from the companies that they would not spin off, sell or close any division or factory. The contract for Delphi workers mirrors GM 's deal; it also allows Delphi employees to transfer back to GM and retire by Jan. 1 with a GM pension. Other terms include a $ 1,350 signing bonus, improved pensions and better cost-of-living adjustments. GM and Delphi workers are expected to complete voting on their new contract by next week. The last national strike by the UAW against Ford was 1976 ; it was also the last time the UAW struck during negotiations. A local union in Atlanta struck Ford in 1986 . The UAW has not held a national strike in years, but has held several local strikes at GM , including a 54-day strike at two Flint plants last summer that shut down most of the company's North American production.