Methods for Semantic Mapping

Roy Gardner

PeaceRep, University of Edinburgh

Contents

1 Introduction

Semantic mapping is a method for finding the semantic relationship between segments in one set of documents – the *source* set – and segments in a later set of documents – the *target* set. Examples include:

- Whether provisions of peace agreements (the source set) are semantically similar to sections of subsequent implementation reports (the target set).
- Whether provisions of peace agreements (the source set) are semantically similar to sections of a subsequent constitution (the target set containing one document).
- Tracking concepts through a sequence of drafts of a document. In this case, any document in the sequence can be the source with the remaining documents acting as the target set.

Figure 1 illustrates some document sequence patterns. The first row is analogous to the agreement–constitution scenario where a set of peace agreements (orange) precede a constitution (in blue). The second row is analogous to the agreement–implementation reports scenario where peace agreements (orange) are followed by implementation reports. The third might be used to analyse the mapping of constitution sections onto later peace agreements.

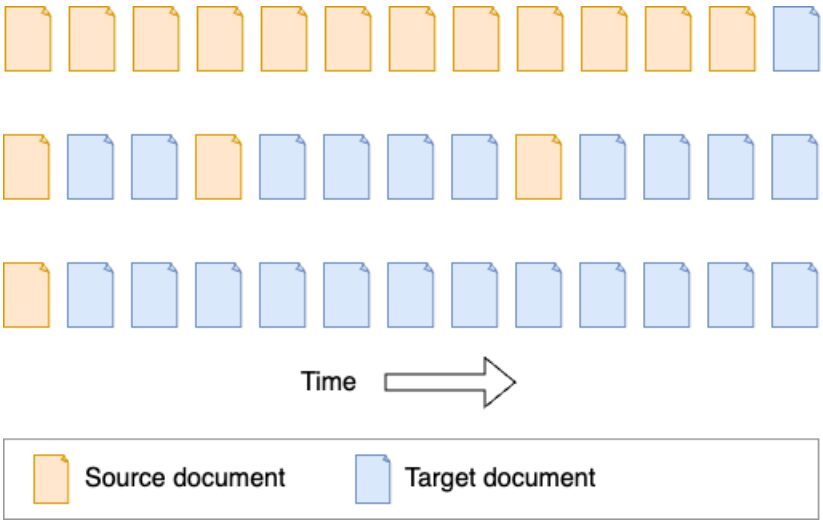

Figure 1: Document sequence patterns.

1.1 Definitions

Source documents A set of documents whose segments are compared to the segments of subsequent set of documents.

Source segments The segments in the source documents.

Target documents A set of documents whose segments are compared to the segments of a previous set of documents.

Target segments The segments in the target documents.

Mapping segments Source segments that are semantically similar to target segments.

Mapped segments Target segments that are semantically similar to source segments.

Semantic Map A map is a data structure that maps keys to values forming key-value pairs. Here a semantic map is a data structure that maps mapping segment identifiers (keys) to sets of mapped segment identifiers (values).

1.2 Measuring Semantic Similarity

Sentence-level semantic similarity measures the degree to which two or more natural language sentences or clauses convey similar meaning. We use version 4 of Google's Universal Sentence Encoder (USE v4)[1][1] to generate 512-length numerical representations of sentences referred to as encoding vectors or embeddings.

The preferred measurement of the distance between a pair of encoding vectors is angular distance. The inverse of this distance is a score of the semantic similarity between the sentences that the vectors represent. Semantic similarity scores range from 1.0 to 0.0 where 1.0 means two sentences are identical in meaning and comprise the same words in the same order. As the meaning of the sentences diverge, the similarity score decreases.

USE models enable efficient and accurate computation of sentence-level encoding vectors, making it possible to perform large-scale semantic similarity tasks

[1] https://www.kaggle.com/models/google/universal-sentence-encoder/frameworks/tensorFlow2/variations/universal-sentence-encoder/versions/2?tfhub-redirect=true

on a range of multi-language datasets without any pre-processing of text other than segmentation and encoding[3][2].

2 Methods

2.1 Document Processing

2.1.1 Segmentation

Document text is segmented into sentence-level segments using the parser component of the spaCy[2] English large language model[3]. Sentence segmentation boundaries are the default punctuation characters defined by spaCy with the addition of semi-colons.

A document is therefore a container for a set of segments. Segment identifiers comprise the ID of the segment's document and an integer value indicating the segment's ordinal position in the document. The complete text of a document can be recreated by combining the segments in order although formatted structure (e.g., headers, lists, etc.) is lost.

Segments inherit the metadata of their document container, for example, peace agreement date, stage, or region. Date is a critical attribute that is used to determine the date order of segments.

2.1.2 Encoding

Encoding vectors are obtained from the USE model for all qualifying text segment. To qualify for encoding, a segment's text must exceed a minimum word count.

2.2 Matrices

2.2.1 Similarity Matrices

A similarity matrix comprises the text segments of the source documents in rows, and the segments of the target documents columns. Cells contain the semantic similarity of a row–column, i.e., source–target, segment pair.

Matrix rows are in the same order as an indexed set of source segment identifiers:

[2] https://spacy.io
[3] https://github.com/explosion/spacy-models/releases/tag/en_core_web_lg-3.7.1

$$R = \{r_1, r_2, \ldots, r_M\} \tag{1}$$

and their encoding vectors

$$\boldsymbol{R} = \{\boldsymbol{r}_1, \boldsymbol{r}_2, \ldots, \boldsymbol{r}_M\} \tag{2}$$

where M is the total number of row segments. The row segments may come from one or more documents.

Matrix columns are in the same order as an indexed set of target text segment identifiers:

$$C = \{c_1, c_2, \ldots, c_N\} \tag{3}$$

and their encoding vectors

$$\boldsymbol{C} = \{\boldsymbol{c}_1, \boldsymbol{c}_2, \ldots, \boldsymbol{c}_N\} \tag{4}$$

where N is the total number of column segments. The column segments may come from one or more documents.

The semantic similarity score $s(r_m, c_n)$ of two text segments (r_m, c_n) is measured as the inverse of the angular distance between the encoding vectors of the two segments.

$$s(r_m, c_n) = 1 - \frac{\arccos\left(\dfrac{\boldsymbol{r}_m \cdot \boldsymbol{c}_n}{||\boldsymbol{r}_m||\,||\boldsymbol{c}_n||}\right)}{\pi} \tag{5}$$

The similarity matrix $\boldsymbol{S}$

$$\boldsymbol{S} = \begin{bmatrix} s(r_1, c_1) & s(r_1, c_2) & \cdots & s(r_1, c_N) \\ s(r_2, c_1) & s(r_2, c_2) & \cdots & s(r_2, c_N) \\ \vdots & \vdots & \ddots & \vdots \\ s(r_M, c_1) & s(r_M, c_2) & \cdots & s(r_M, c_N) \end{bmatrix}$$

is generated by computing the semantic similarity of every pair of source and target segments.

2.2.2 Date Matrix

A date matrix $\boldsymbol{D}$ is a binary-valued matrix with the same dimensions as a similarity matrix $\boldsymbol{S}$, and with rows and columns in the same order as the R and C segment sets respectively. The function of the date matrix is to ensure that mapping operates forwards in time.

The value of a cell in $\boldsymbol{D}$ is:

$$d_{m,n} = \begin{cases} 0 & if\ date_{r_m} \geq date_{c_n} \\ 1 & if\ date_{r_m} < date_{c_n} \end{cases} \tag{6}$$

Figure 2 is a schematic diagram of a date matrix for two source documents in rows (S1 and S2), and three target documents (T1, T2, and T3) in columns. S1 contains 10 segments and has 10 rows in the matrix. S2 has six segments and hence six rows in the matrix. The segment counts for T1, T2, and T3 are 12, 7, and 9 respectively. The document time sequence is S1,T1,T2,S2,T3, i.e., $date_{S1} < date_{T1} < date_{T2} < date_{S2} < date_{T2}$.

All matrix cells in the rows of S1 contain the value 1 because the dates of S1 segments are earlier than the dates of the segments of all three target documents. The cells in the rows of S2, on the other hand, contain the value 0 for the segments of T1 and T2 because both T1 and T2 precede S2. The only cells in the S2 rows that contain the value 1 are those for the segments of the later T3 document.

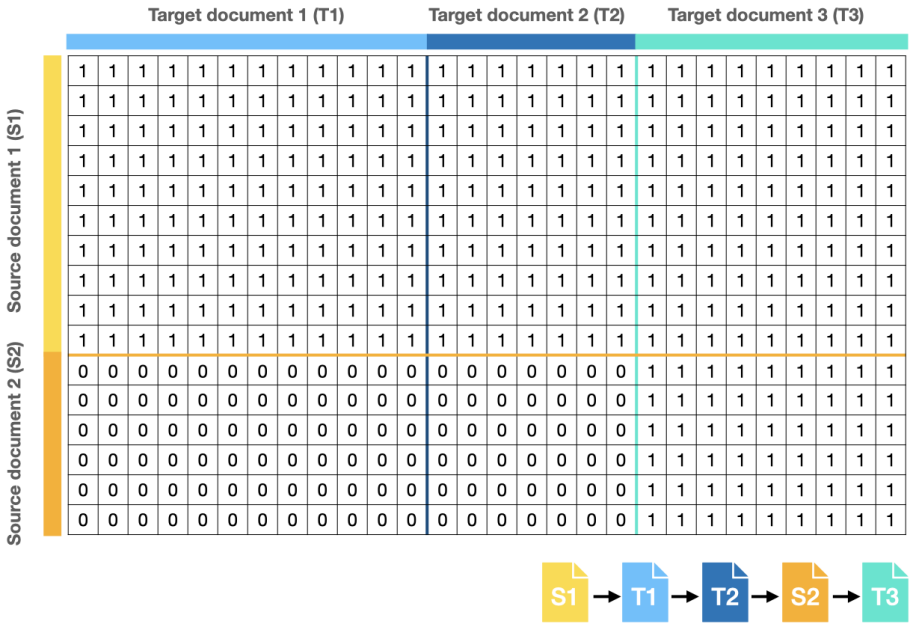

Figure 2: Schematic diagram of a date matrix.

2.2.3 Thresholded Similarity Matrices

A similarity matrix is a dense matrix with a non-zero semantic similarity score in every cell. In many cases, pairs of segments are not semantically similar and therefore, the vast majority of cell values are low. To remove these low value pairings, and prepare the matrix for further processing, the similarity matrix is thresholded.

A threshold θ is applied to convert a similarity matrix $\boldsymbol{S}$ into a thresholded matrix $\boldsymbol{U}$ as follows:

$$u_{m,n} = \begin{cases} 0 & if\ s_{m,n} < \theta \\ u_{m,n} & if\ s_{m,n} \geq \theta \end{cases} \tag{7}$$

2.3 Semantic Mapping

The semantic mapping process is as follows:

1. Generate a semantic similarity matrix $\boldsymbol{S}$, and a date matrix $\boldsymbol{D}$ for a set of source and target segments.

2. Compute the thresholded matrix $\boldsymbol{U}$.
3. Compute $\boldsymbol{V}$, the element-wise product of the thresholded matrix $\boldsymbol{U}$ and the date matrix $\boldsymbol{D}$:

$$\boldsymbol{V} = \boldsymbol{U} \odot \boldsymbol{D} \tag{8}$$

The operation in Equation 8 ensures that elements in a row vector of $\boldsymbol{V}$ only contain non-zero values when the date of a target (column) segment is later than the date of the row's segment.

A row in $\boldsymbol{V}$ contains a *mapping* segment if the row vector contains at least one non-zero element. Furthermore, any non-zero elements in a row vector correspond to *mapped* segments with a date later than the date of the *mapping* segment in the row. The count of non-zero values in a row vector is the number of target segments mapped by the row's segment.

Figure 3 is a schematic of a thresholded matrix with the rows of mapping segments highlighted. The use of the date matrix (Equation 8) ensures that S2 segments cannot map onto T1 and T2 segments.

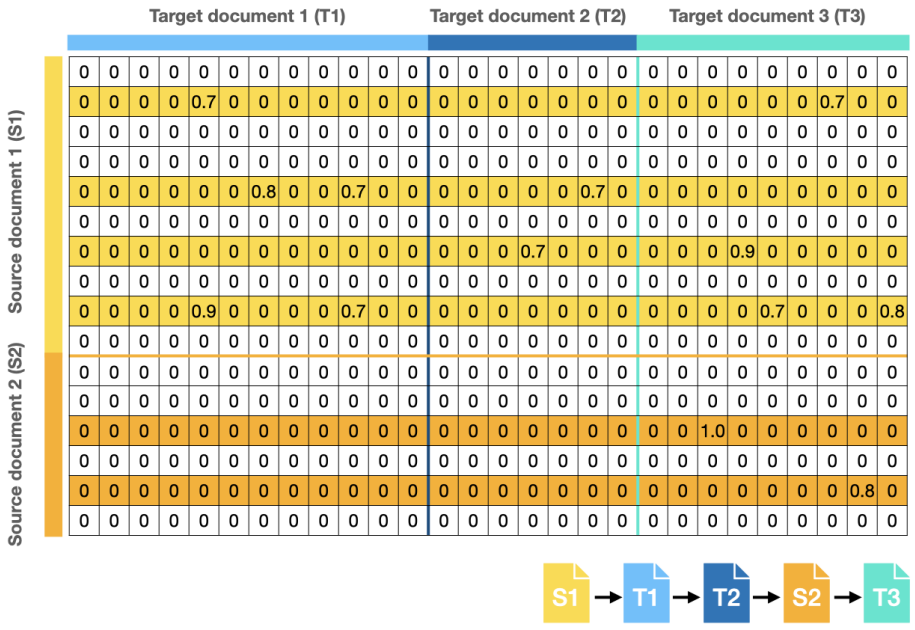

Figure 3: Row vectors in thresholded matrix.

A column in $\boldsymbol{V}$ contains a *mapped* segment if the column vector contains at least one non-zero element. The count of non-zero values in a column vector is the number of times the column's target segment is mapped by source segments.

Figure 4 is a schematic of a thresholded matrix with the columns of mapped target segments highlighted. The use of the date matrix (Equation 8) ensures that T1 and T2 segments cannot be mapped by S2 segments.

Figure 4: Column vectors in thresholded matrix.

Finally, a semantic map is constructed where the keys are *mapping* segment identifiers, and the values are set of tuples M where each tuple contains a *mapped* segment identifier and the semantic similarity score between the *mapping* and *mapped* segments:

$$r_m \mapsto M_{r_m} \tag{9}$$

where:

$$(c_n, \boldsymbol{v}_{m,n}) \begin{cases} \in M_{r_m} & if\ \boldsymbol{v}_{m,n} > 0 \\ \notin M_{r_m} & if\ \boldsymbol{v}_{m,n} = 0 \end{cases} \tag{10}$$

A semantic map is the foundational data structure for semantic mapping anal-

ysis – see Demonstators section below. Examples include:

- Counting the number of target segments mapped by the segments of a source document.
- Organising source and target documents in time order.
- Analysis by source and target document metadata attributes.

3 Demonstrators

3.1 Agreement–Constitution Mapping

The agreement–constitution demonstrator[4] displays semantic map analysis for a number of countries where a set of peace agreements precede a constitution. In addition to navigable trees for exploring the relationship between peace agreement provisions and constitution sections, sample constitution sections, timelines, and metadata analysis are also available. Constitution sections are used to deep link into the Constitute Project website.

3.2 Agreement–Implementation Report Mapping

The agreement–implementation demonstrator[5] contains a navigable tree for exploring the relationship between peace agreement provisions and implementation report sections for the Philippines, South Sudan, and Ukraine.

A variant of the methodology can be viewed for South Sudan and Ukraine. In these examples, agreement provisions are pre-selected on the basis of their semantic similarity to topics (short text segments that capture the intent of PA-X codebook codes).

4 Summary

This paper describes a matrix-based method for exploring the semantic relationship between two sets of documents where chronology is preserved. The method has been applied to peace process document sets (agreements, constitutions, and implementation reports). There are links to interactive demonstrators to illustrate these applications of the methodology.

5 References

[1] Daniel Cer, Yinfei Yang, Sheng yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. Universal sentence encoder, 2018.

[2] Cruz A, Elkins Z, Gardner R, Martin M, Moran A. A new method for analyzing public consultation data. *PLoS ONE*, 18(12): e0295396, 2023.

[3] Roy Gardner. Semantic Analysis to Support Peace Analytics. *Peace Analytics Series. PeaceRep: The Peace and Conflict Resolution Evidence Platform, University of Edinburgh*, 2023.

[4] Roy Gardner. Peace Agreements-Constitutions Semantic Mapping Demonstrator. https://peacerep.github.io/agreement-constitution-mapping/index.html, 2024.

[5] Roy Gardner. Peace Agreements-Implementation Reports Semantic Mapping Demonstrator. https://peacerep.github.io/agreement-report-mapping/index.html, 2024.

[6] Roy Gardner. Topic Discovery, Summarisation, and Diffusion. *PeaceRep Lab Technical Report*, 2024.