1 Definition

Similarity by cosine determines the degree of similarity between two sentences by calculating the cosine of the angle formed by the two vectors in each sentence.
Consider the following sentences :

1. Bonjour John. [1]
2. Bonjour Doé. [2]

Note that the two sentences are similar, but how can you tell with a computer ?

2 Simple 2-dimensional space example

Consider the following sentences :

1. Bonjour John.
2. Bonjour. [3]

Note that one of the two sentences is made up of two words, so they can be represented in a two-dimensional space.
The following similarity table is created :

Sentences	**Bonjour**	**John**
Bonjour John	1	1
Bonjour	1	0

TABLE 1 – Simple similarity table example 1

The representation in a reference frame gives :

1. 'Hello John.' in French
2. 'Hello Doé.' in French
3. 'Hello.' in French

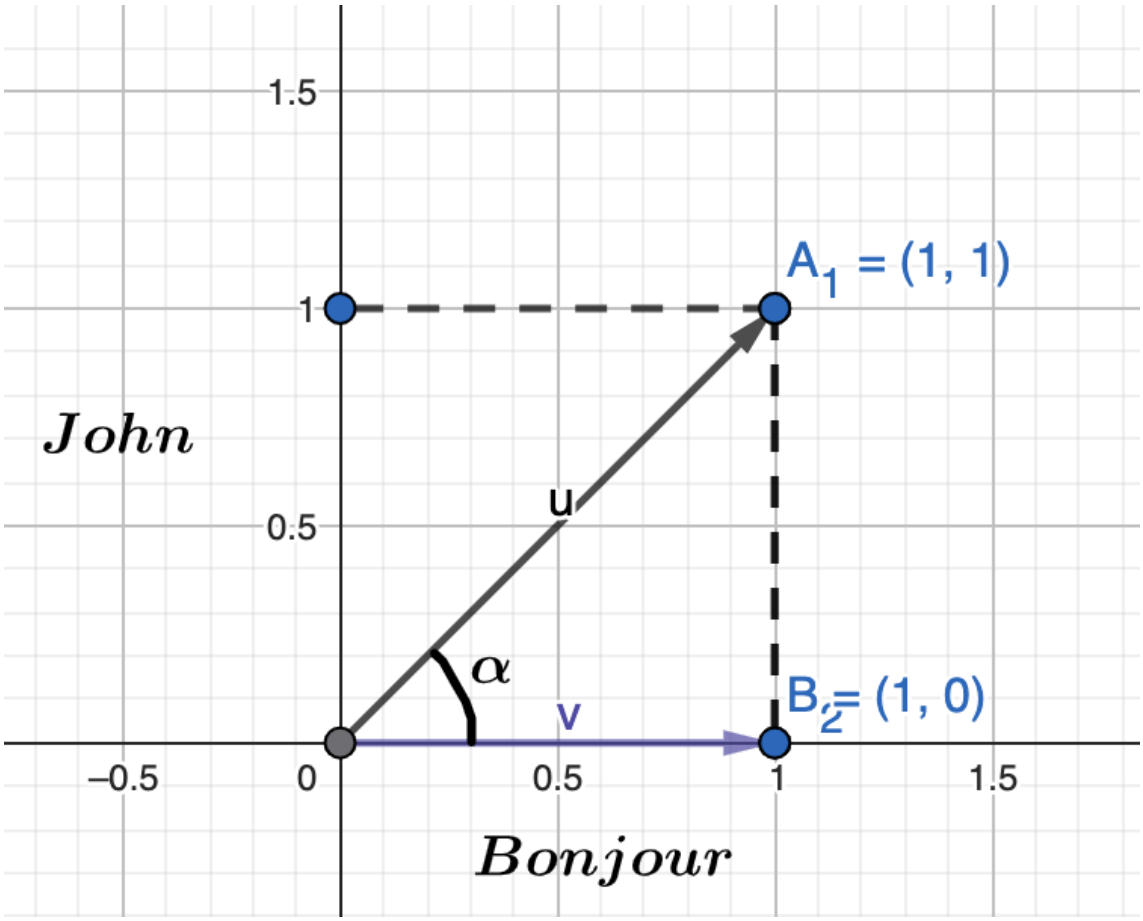

FIGURE 1 – Simple representation similarity example 1

Sentence 1 is represented by the vector $A_1(1,1)$ and sentence 2 by the vector $B_2(1,0)$.
The angle formed by these two vectors is the α angle, whose cosine is used to calculate the similarity between the two sentences.
In the previous example, $\alpha = 45^o$ of which $cos(\alpha) = 0.71$. So 71% chance that the two sentences are similar.
Suppose sentence 2 is **Bonjour, Bonjour, Bonjour.** then we get :

Sentences	**Bonjour**	**John**
Bonjour John	1	1
Bonjour, Bonjour, Bonjour	3	0

TABLE 2 – Simple similarity table example 2

The representation in a reference frame gives :

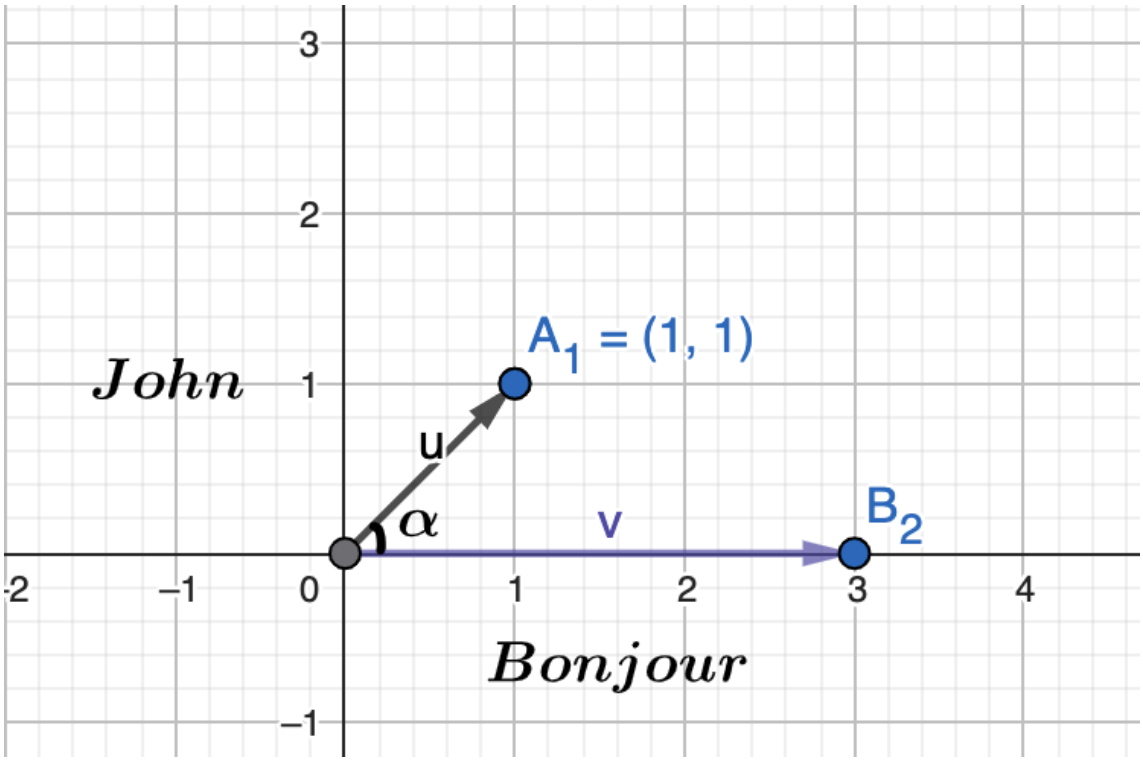

FIGURE 2 – Simple representation similarity example 2

Regardless of the length of sentence 2, the number of times *Bonjour* is added does not change the cosine of α.

Summary : Cosine similarity is the cosine of the angle between two vectors, revealing the similarity between them. Its value is always between 0 and 1.

3 Case of exact similarity

Consider the following two sentences :

1. Bonjour John
2. Bonjour John

It's like a table of similarities :

Sentences	Bonjour	John
Bonjour John	1	1
Bonjour John	1	1

TABLE 3 – Exact similarity table

The representation in a reference frame gives :

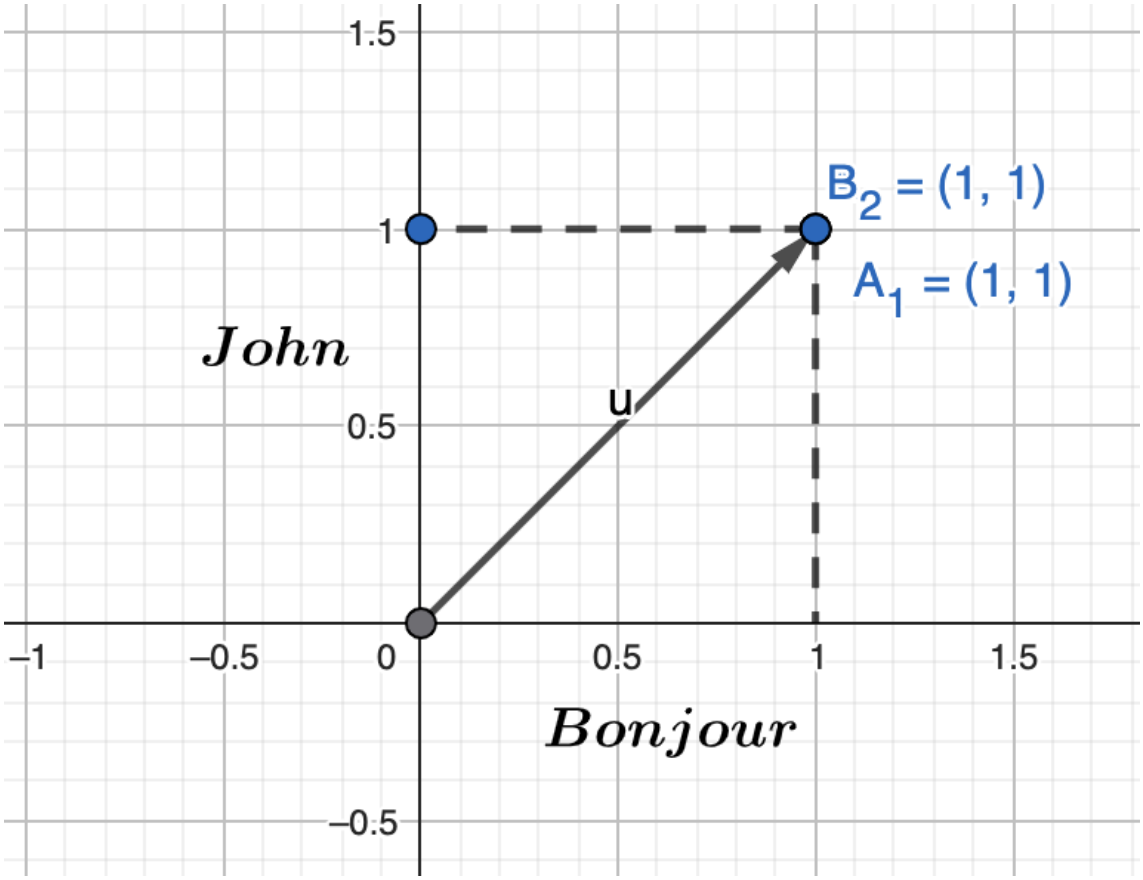

FIGURE 3 – Exact similarity representation

The two vectors coincide, so $\alpha = 0^o$ or $cos(\alpha) = 1$, i.e. 100% chance that the two vectors are similar. We can conclude that the two sentences are totally similar.

4 No similarity

Consider the following two sentences :

1. Bonjour.
2. John.

The similarity table gives :

Sentences	Bonjour	John
Bonjour	1	0
John	0	1

TABLE 4 – Table no similarities

The representation in a reference frame gives :

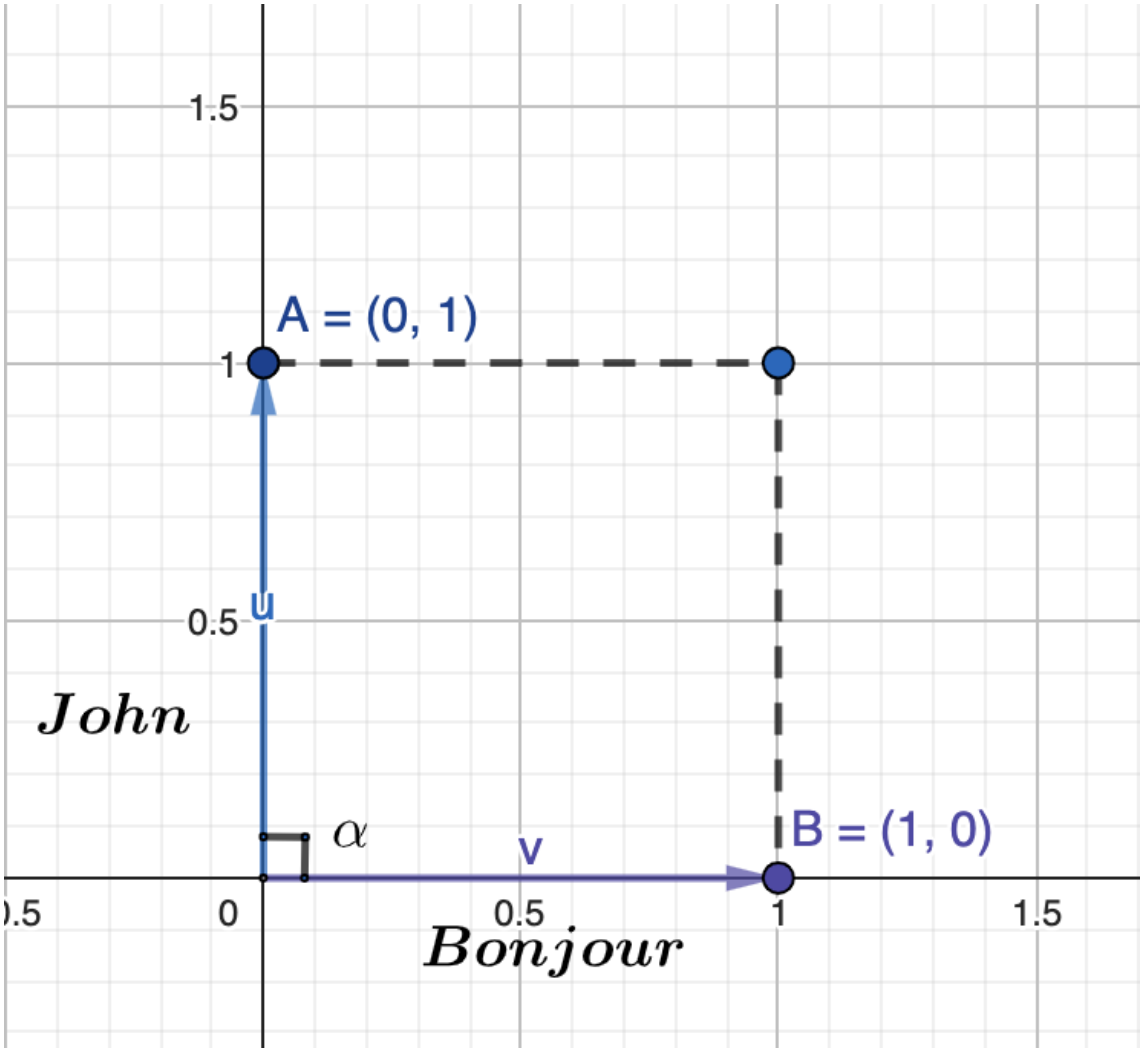

FIGURE 4 – Non-similarity representation

The angle formed by the two vectors is $\alpha = 90^o$ so $cos(\alpha) = 0$ or a 0% chance that the two vectors are similar. We can conclude that the two sentences are completely different.

5 Summary

$$cos(\alpha) = \begin{cases} 0 & \text{If there is no similarity between the two sentences.} \\ 1 & \text{Exact similarity.} \\ \in]0, 1[& \text{Sensibly similar when both sentences have words in common.} \end{cases}$$

To get the **cosinus** of the similarity between two sentences, follow these steps :

1. Make a word frequency chart (count the number of occurrences of each word in each sentence).
2. Show points.
3. Find the angle between Vectors.
4. Calculate the cosine of the formed angle.

6 Complex cases

The previous cases work for sentences with two words, i.e. representable in a 2-dimensional space ℝxℝ.
Real-life sentences are either multi-word or representable in a $\mathbb{R}^n$ space, in which case we use the following formula :

$$cos(\alpha) = \frac{\sum_{i=1}^{n} A_i Bi}{\sqrt{\sum_{i=1}^{n} A_i^2} * \sqrt{\sum_{i=1}^{n} B_i^2}}, avec\ i = \text{ word index}$$

Consider the following sentences :

1. Bonjour John
2. Bonjour

Sentences	**Bonjour**	**John**
Bonjour John	1	1
Bonjour	1	0

Table 5 – Simple similarity table example 1

Calculating the similarity gives :

$$cos(\alpha) = \frac{(1*1)+(1*0)}{\sqrt{1^2+1^2} * \sqrt{1^2+0^2}} = \frac{1}{\sqrt{2}*1} = 0.7071$$

The $cos(45)$ found previously.
Let's consider two sentences :

1. Bonjour tout le monde.
2. Bonjour John

The similarity table gives :

Sentences	**Bonjour**	**tout**	**le**	**monde**	**John**
Bonjour tout le monde.	1	1	1	1	0
Bonjour John	1	0	0	0	1

Table 6 – Multidimensional similarity table

Calculating the similarity gives :

$$cos(\alpha) = \frac{(1*1)+(1*0)+(1*0)+(1*0)+(0*1)}{\sqrt{1^2+1^2+1^2+1^2+0^2} * \sqrt{1^2+0^2+0^2+0^2+1^2}} = \frac{1}{2*\sqrt{2}} = 0.35$$

A 35% chance of being similar.

7 Go further

The problem with this method is that similarity is based on word construction (syntactic/grammatical level), so two words **Hi** and **Hello** will be considered dissimilar because the implicit meaning is not taken into account.
Similarly, **better** and **good** will be considered as not similar..
In projects, we use the word lemma, i.e. the basic form of the word, to calculate similarity. Using the lemma, we get :

- better => lemma good
- good => lemma good

Using the lemmas **better** et **good** will be represented by their lemmatized form and therefore similar.
Furthermore, this method does not take into account the semantic meaning between two sentences, i.e. two sentences **The cat eats the mouse** and **The mouse eats the cat** will be considered syntactically and grammatically similar, but semantically (meaning-wise) they are totally different.
To take the subject further, you can use NLP (Natural Language Processing) libraries such as SpaCy and NLTK, which contain automatic language processing tools.
Code examples can be found here : Calculating similarity in different ways which shows various calculation methods natively with Python, SpaCy, NLTK.