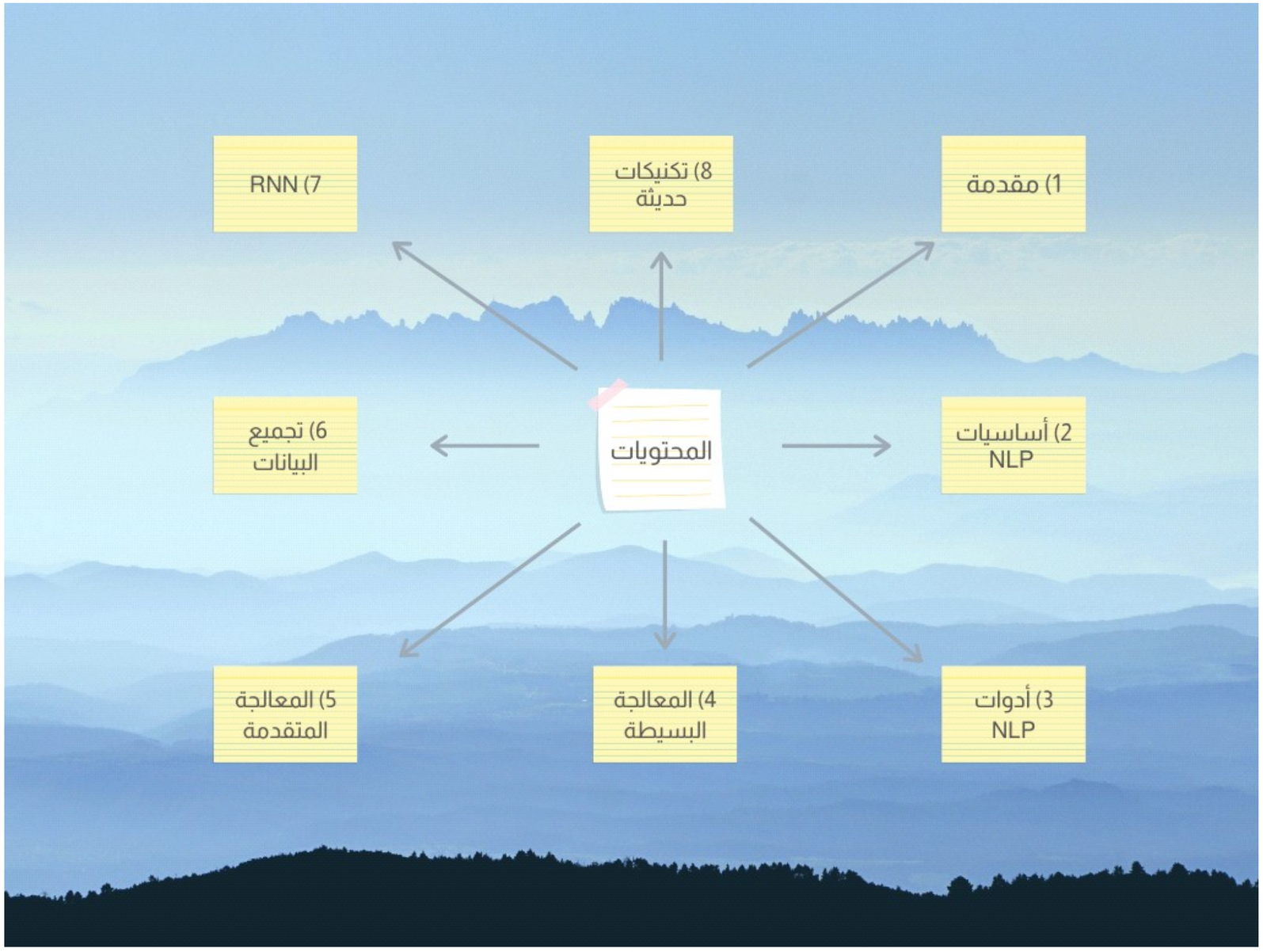
Natural Language Processing المعالجة اللغوية الطبيعية



المحتويات

1) مقدمة	المحتويات	ما هو NLP	تاريخ NLP	العقبات و التحديات	التطبيقات				
2) أساسيات NLP	المكتبات	الملفات النصية	ملفات pdf	البحث في النصوص					
3) أدوات NLP	Tokenization	Sent. Segm.	POS	Stem & Lemm	NER	Stopwords	Matchers	Syntactic Struc.	T.Visualization
4)المعالجة البسيطة	Word Meaning	Word embed	T. Vectors	Word2Vec	BOW	TF-IDF	Text Similarity	Dist. Similarity	
5)المعاجلة المتقدمة	T. Classification	T. Clustering	LDA	NMF	L. Modeling	GloVe	Lexicons	NGrams	T. Generation
	Sent. Analysis	Naïve Bayes	Vader	Auto Correct	Ans. Questions			Summarization & Snippets	
6)تجميع البيانات	Tweet Collecting	Data Scraping	Information Extraction		Information Retrieval			Relative Extraction	Search Engine
7) RNN	Seq to Seq	LSTM	GRU	Rec NN\TNN					
8)تكنيكات حديثة	Word Cloud	CNN	T. Forcing	Attention Model	Transformer	Bert	FastText	Gensim	Chat Bot

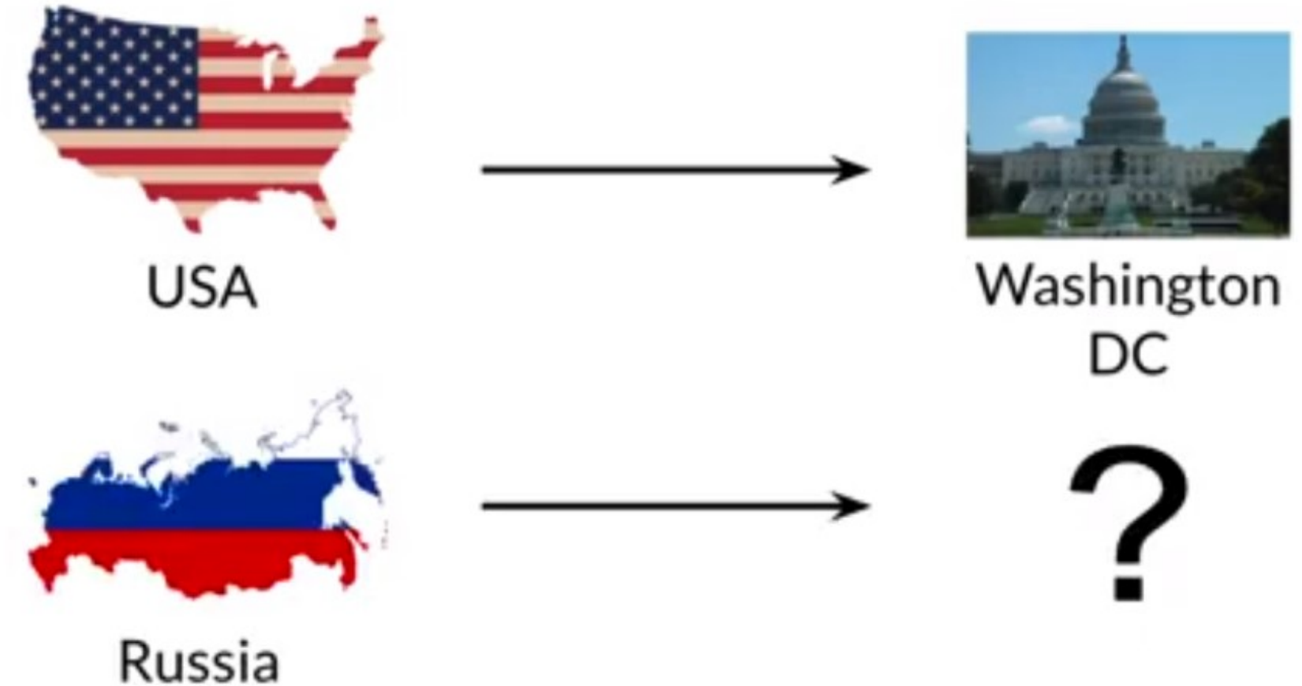
القسم الرابع : المعالجة البسيطة للنصوص

الجزء الثالث : Text Vectors

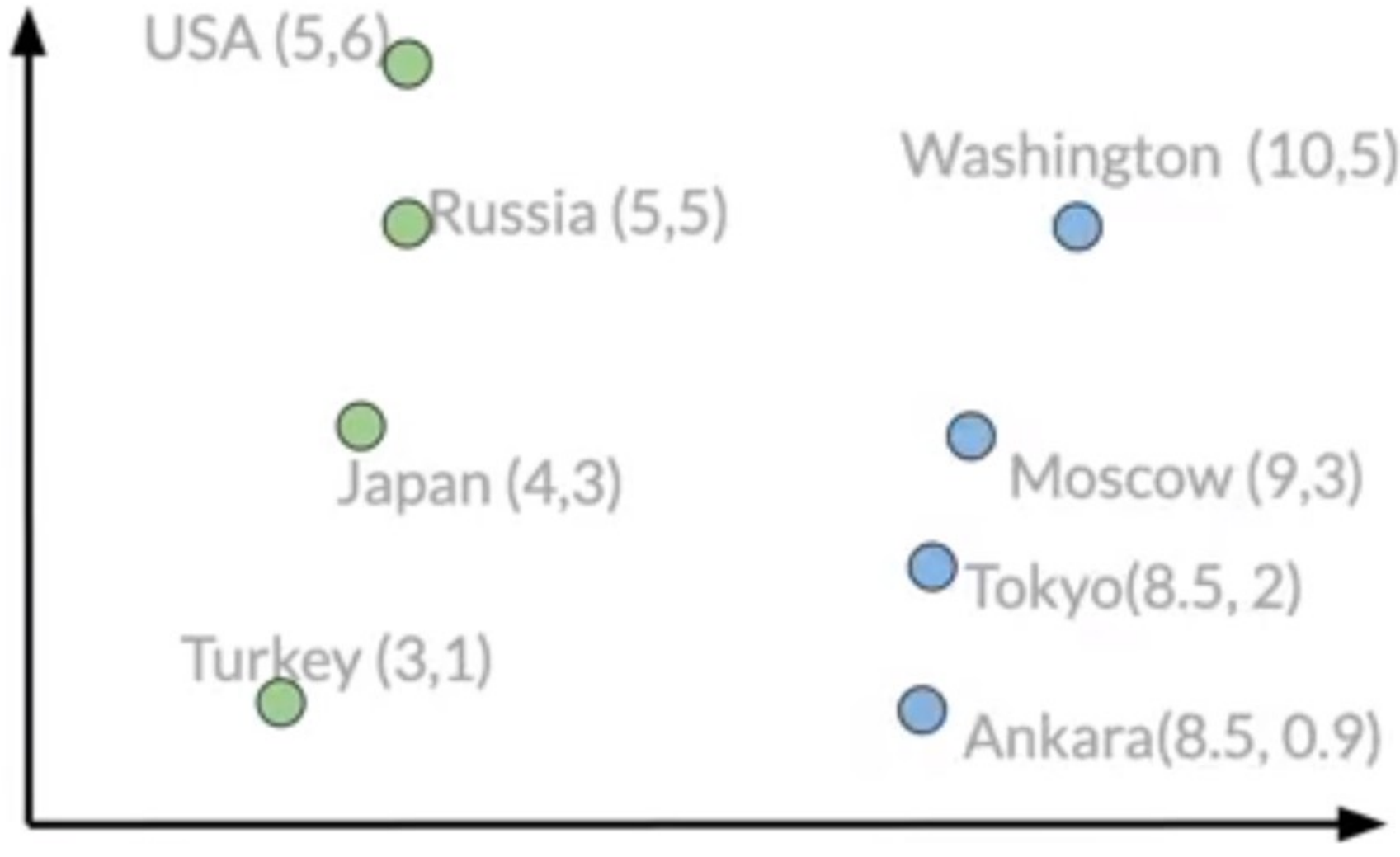
==========

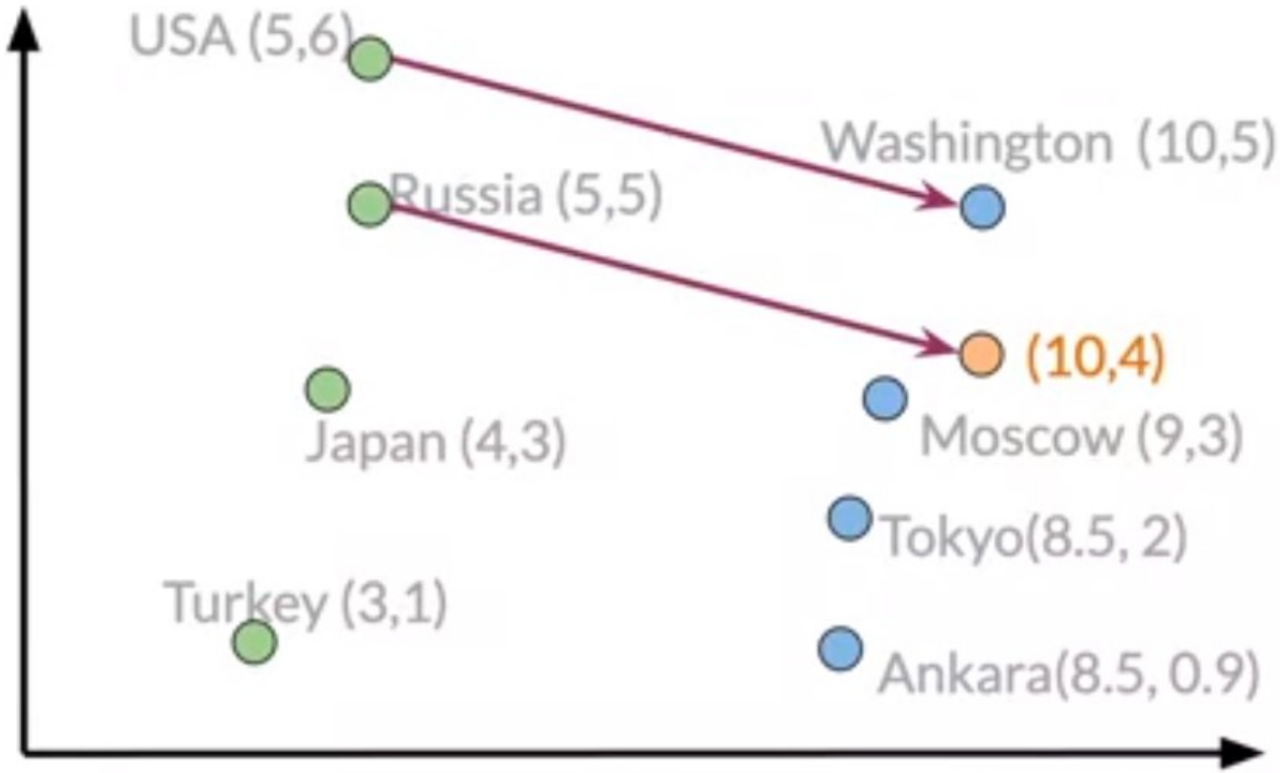
نتناول في هذا الجزء , العمليات الحسابية المعتمدة علي قيم word embedding لكل كلمة , وذلك لتحديد كلمات طلوبة , بناء علي عدد آخر من الكلمات . .

مثال بسيط : لو كان لدينا عاصمة الولايات المتحدة و نريد استنتاج عاصمة روسيا , فكيف يتم هذا ؟



يتم عبر رسم الدول و العواصم في جراف معين , فلو تخيلنا ان قيم WE هي قيمتين فقط مثل هذا الرسم , وتم تحديد اماكن الدول و المدن الهمة هنا , و تم تطبيق عملية طرح بين امريكا و عاصمتها , التي ستكون 1- , 5 ثم مع هذا الفارق مع قيم روسيا , والتي ستكون بقيمة معينة 10,4





Washington - USA = $\begin{bmatrix} 5 & -1 \end{bmatrix}$

Russia + $\begin{bmatrix} 5 & -1 \end{bmatrix}$ = $\begin{bmatrix} 10 & 4 \end{bmatrix}$

و بالبحث , نجد ان اقرب مدينة لهذه القيمة هو موسكو , و يمكن تطبيق هذا الامر علي عدد اخر من البلاد

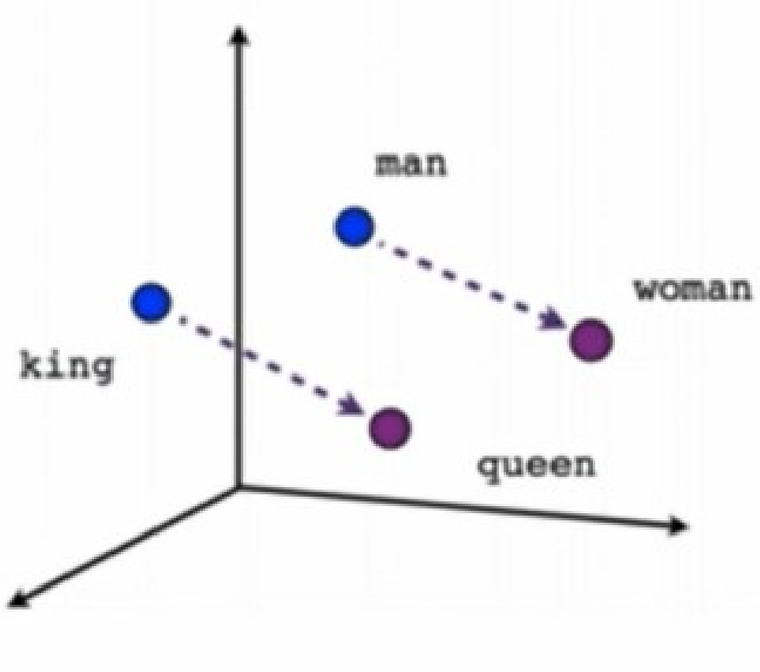
و هذا الأمر لو تم علي مقياس صغير مثل بعدين فقط , فيمكن أن يتم علي حجم أكبر , مثل 300 بعد الخاص بالـ WE

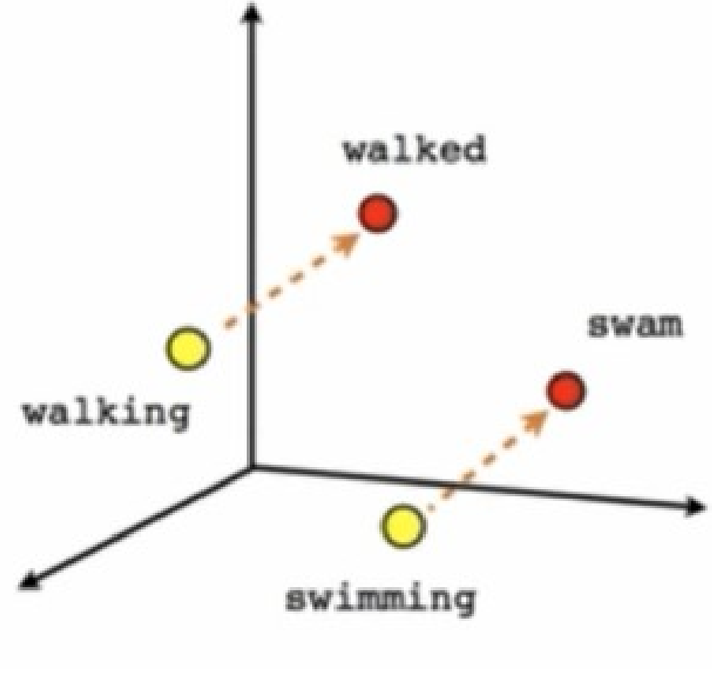
كذلك يمكن عمل جمع او طرح للجملة المشهورة, فيمكن مثلا ان نقول ان :

X = King – Man + Woman

قيمة x هنا هي ان نقوم بايجاد الفارق بين الملك و الرجل , والتي ستكون صفات الملوك , واذا اضفناها الي المراة , ستكون هي قيمة الملكة . . و عمليات الجمع و الطرح هي في الحقيقة لكل الفيكتورز الـ 300

كذلك نفس الامر في الازمنة في الافعال وغيرها هكذا





و هذا الأمر يسمي word analogy او تشابه الكلمات

و هي الخاصة بمدي اقتراب او ابتعاد الكلمات عن بعضها البعض ,و مدي الاختلافات او التشابهات بينها , وهناك امثلة اخري مثل :

King - Queen ~= Prince - Princess

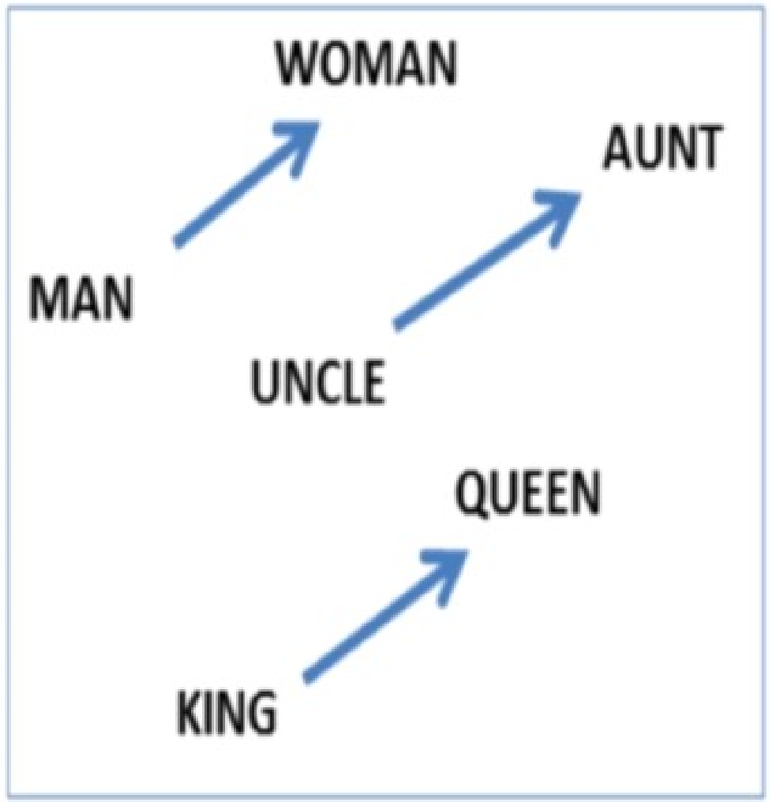
France - Paris ~= Germany - Berlin

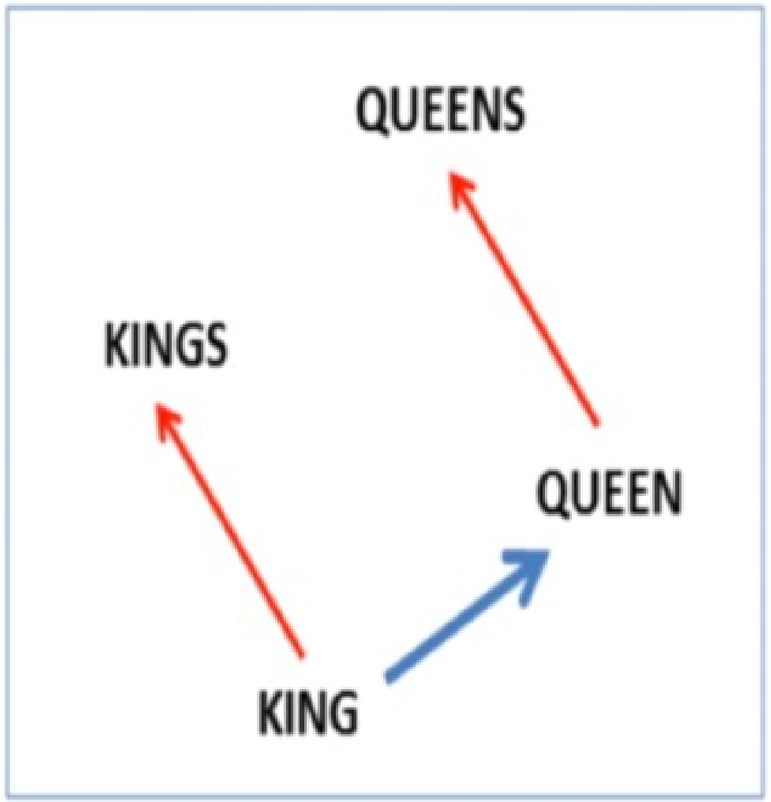
Japan - Japanese ~= China - Chinese

Brother - Sister ~= Uncle - Aunt

Walk - Walking ~= Swim - Swimming

و التي تكون بهذا الشكل





ففكرة طرح كلمات من بعضها , او مساواة الفروق ببعضها , تعني أن الفرق بين الكلمة الاولي و الثانية هو متجه , مساوي في القيمة و الاتجاه , من المتجه الذي يمثل الفارق بين الكلمتين الثالثة و الرابعة

و من الأمور الهامة في التعامل مع متجهات النصوص , هي أن العين البشرية تعجز عن رؤية أكثر من 3 متجهات (3 ابعاد) بينما في الواقع قد يكون هناك عشرات المتجهات

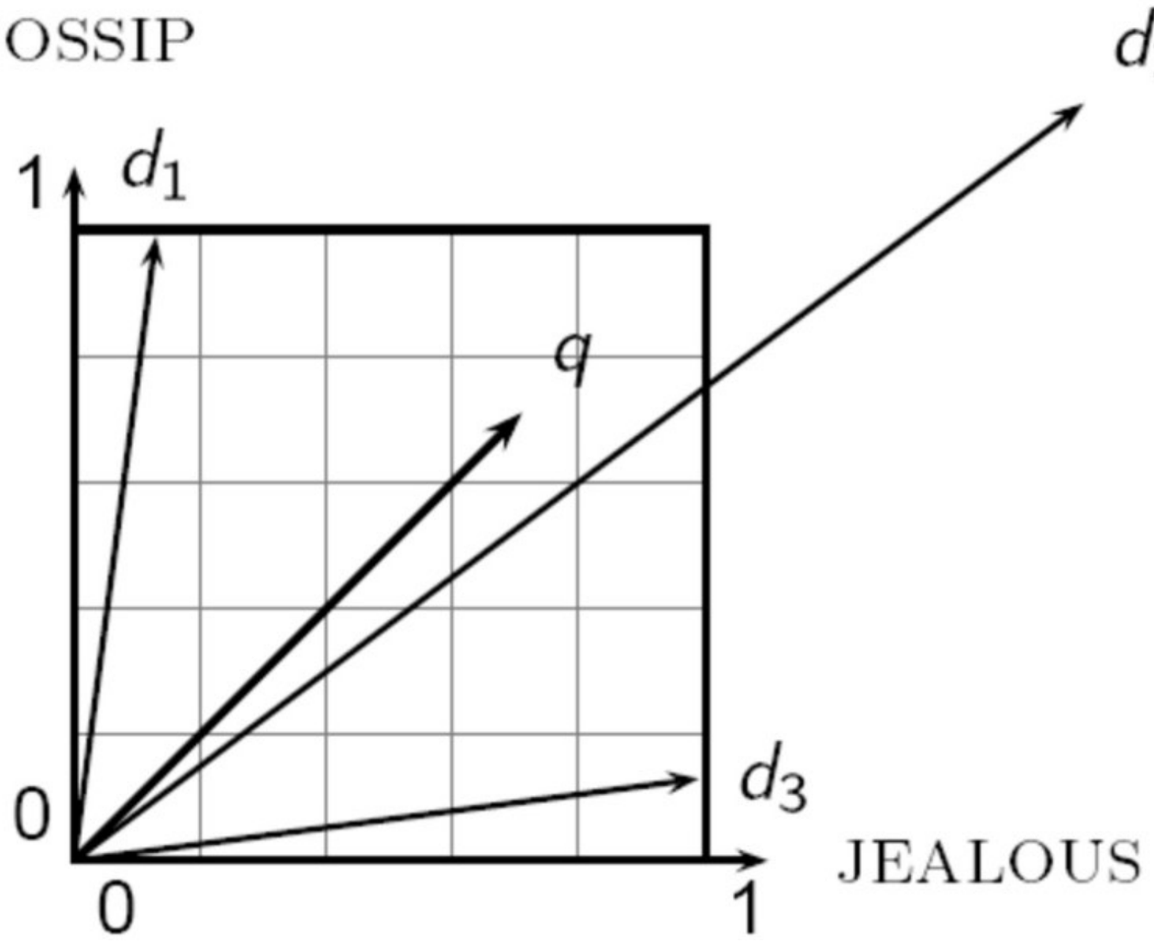
لذا فأحيانا نقوم باستخدام أحد أدوات "تقليل الأبعاد" مثل PCA او TSNE , والتي يكون لها دور في تقليل الابعاد لرسمها

و من تطبيقاته Glove كما سنري لاحقا

_

الان , كيف يمكن حساب مدي اقتراب او ابتعاد كلمتين عن بعضهما البعض ؟ ؟

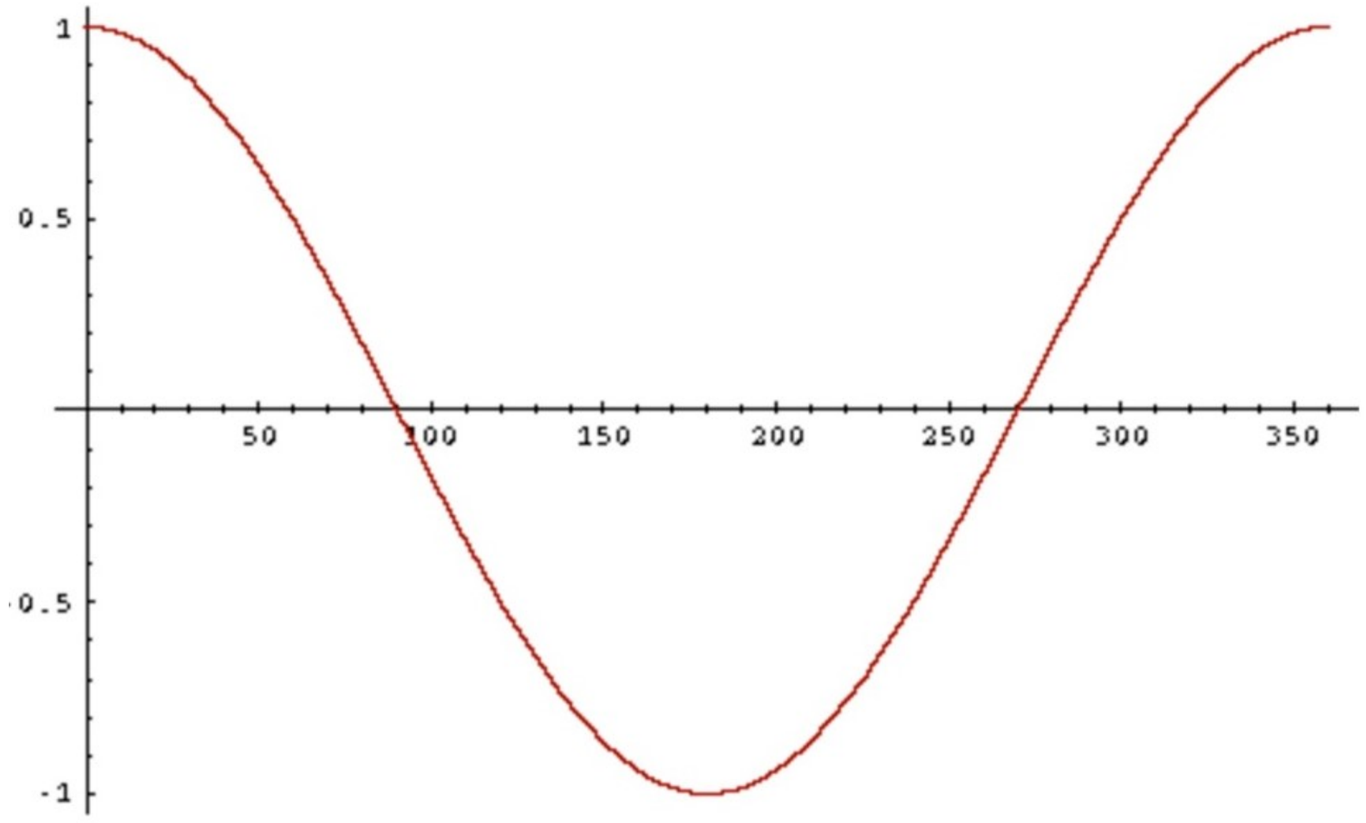
التناول الأول هو المسافة بأسلوب Euclidean distance , وهي أن يتم حساب المسافة بين النقطتين بعضها , بحيث تكون المسافة بين كل متجه و الثاني , يدل علي العلاقة بين الكلمتين هكذا:



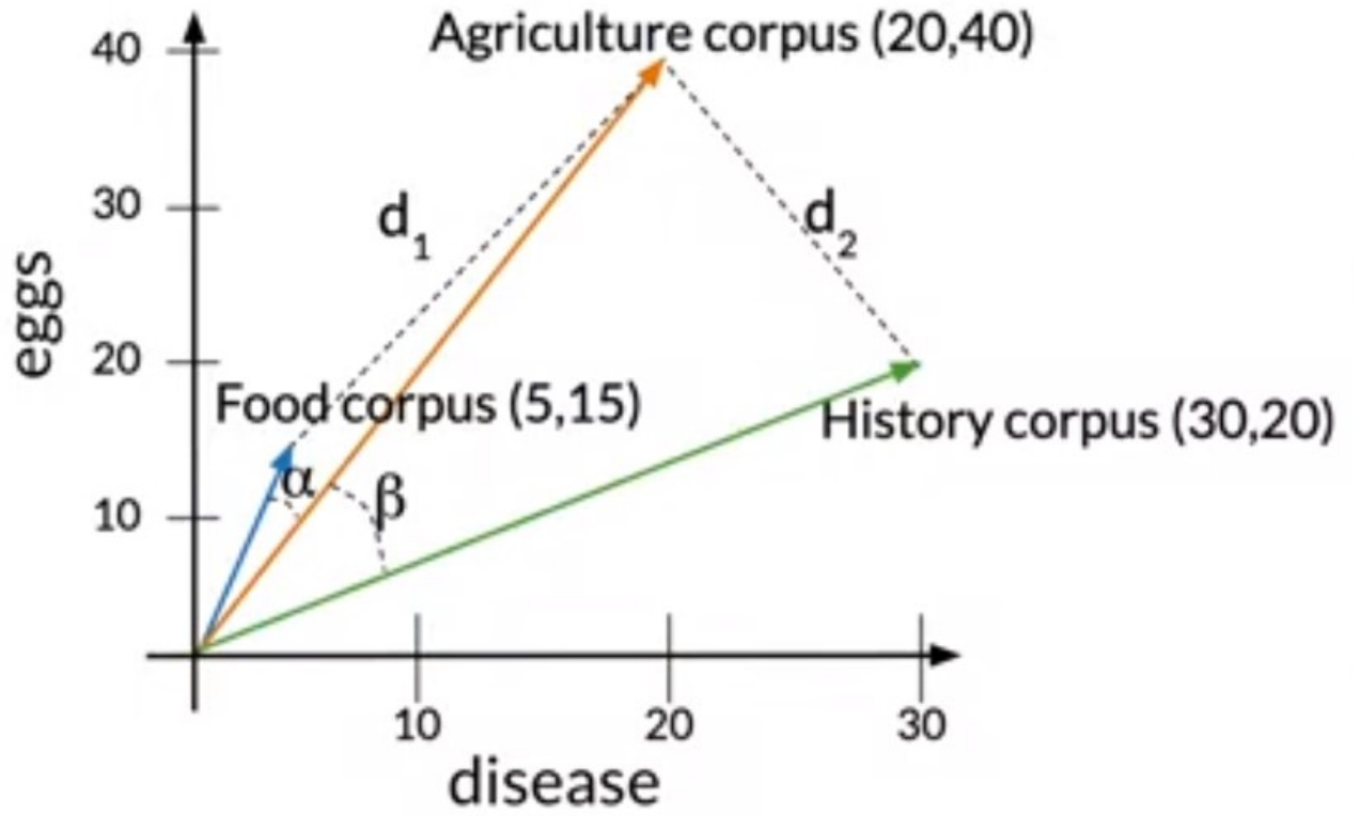
لكن المشكلة ان المسافة بين نقطتين لا يعبر عن الفرق الحقيقي بينهما , لأن المسافة بين q و d2 اكبر من مسافتها مع d1 او d3 , علي الرغم من ان التطابق يكاد يكون بينها و بين d2

لذا فاننا نعتمد علي الزاوية بينهم , و بالتحديد قيمة cos الزاوية فيما يسمي cosine similarity

حيث أن الكلمتين لو كانا متطابقتين فتكون الزاوية بينهم 0 فتكون القيمة 1 , وكلما ابتعدا لما زادت قيمة cos حتي تصل ل 0 مع زاوية 90 و سالب 1 مع الزاوية العكسية , فيكون الترتيب تنازليا حسب المسافة , وتصاعديا حسب قيمة cos هكذا :



و هنا مثال آخر للفارق بين Euclidean distance و cosine similarity



Euclidean distance: $d_2 < d_1$

Angles comparison: $\beta > \alpha$

The cosine of the angle between the vectors

_

اذن كيف يتم حساب قيمة cosine similarity خاصة اننا ليس لدينا قيمة للزاوية بين الكلمات بالفعل

اولا نتذكر معلومة هامة و هي vector norm & dot products و التي يتم حسابها هكذا

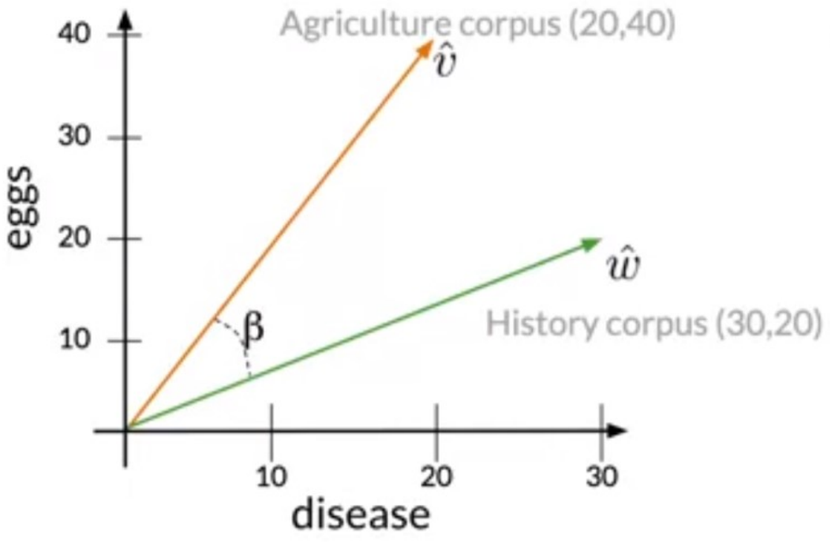
Vector norm

$$\|\vec{v}\| = \sqrt{\sum_{i=1}^{n} v_i^2}$$

Dot product

$$\vec{v}.\vec{w} = \sum_{i=1}^{n} v_i.w_i$$

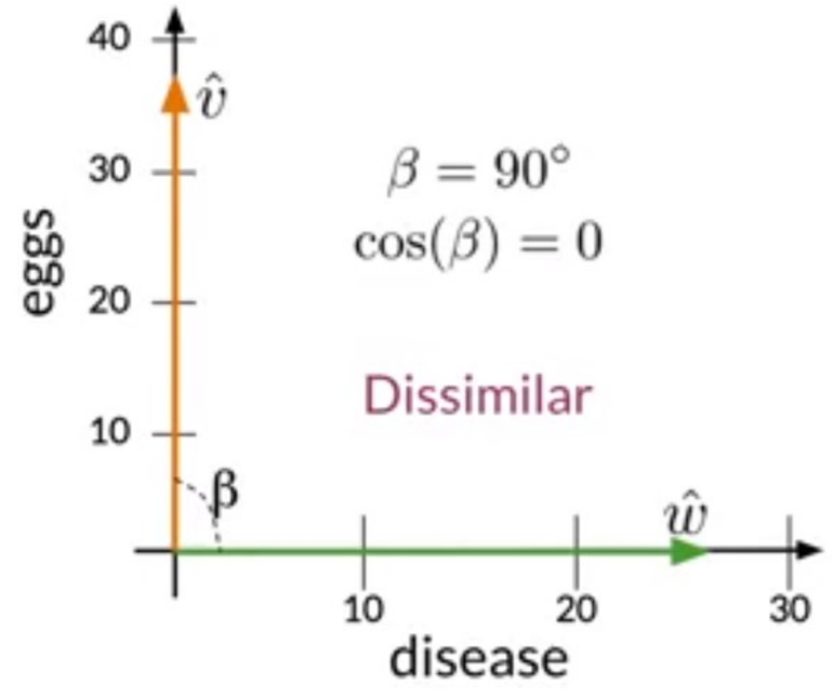
و نستخدم هذه القاعدة المعروفة لحساب قيمة ال cos

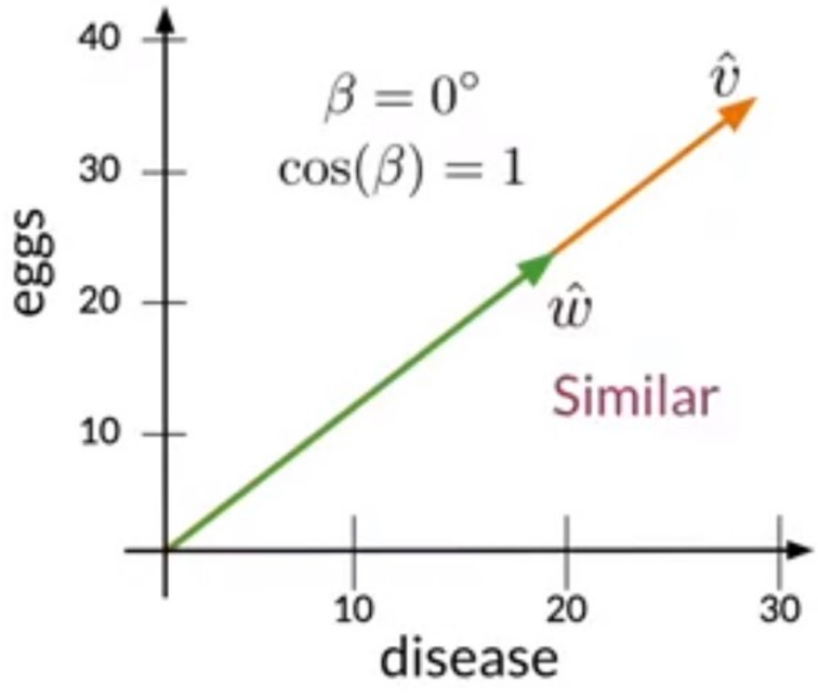


$$\hat{v} \cdot \hat{w} = \|\hat{v}\| \|\hat{w}\| \cos(\beta)$$

$$\cos(\beta) = \frac{\hat{v} \cdot \hat{w}}{\|\hat{v}\| \|\hat{w}\|}$$

و لا تنس ان لو تعاملنا مع كلمتين بعيدتين عن بعضهما فتكون الزاوية 90 و يكون ال cos يساوي 0 اي مختلفين
و لو تعاملنا مع كلمتين متطابقين عن بعضهما فتكون الزاوية 0 و يكون ال cos يساوي1 اي متشابهين
و اي رقم بينهما يشير الي مدي التشابه بينهم





و لن تزيد الزاوية بينهما عن 90 او تقترب من 180 لان هذا يشير الي تضاد و لا توجد قيم سالبة في عدد الكلمات