## 1. Text Rank:

Experiments with several various co-occurrence window widths ranging from 2 to 10 were conducted in this technique, with 2 providing comparatively better results. In addition, to reduce noise, they use syntactic filters to remove just Nouns and Adjectives as probable candidate nodes when creating the graphs.

Once the graph is created, they use Page Rank until convergence to rank each node in the graph. The unweighted-undirected form of the pattern recognition algorithm is shown in the equation below —

Here, d is known as the Damping Factor (it is set to 0.85 to ensure that PR does not get trapped in graph cycles and may readily "teleport" to another node in the network). In(V) is the in-degree of node V, Out(V) is the out-degree of node V, and S(V) is the Page rank score for any given node. The graphic below depicts the Page Rank computation for a node in the graph using the previously given equation. It is worth noting that, since the graph is undirected, In-degree==Out-degree for every node in the graph.

Following convergence, each node in the network is assigned a numeric score that reflects its PageRank(PR) score. All of the keywords that initially exist as a neighbor in the real texts are combined to produce a single keyword as part of the post-processing stage to extract multi-word phrases as well.

## 2. Expand Rank:

In Expand Rank algorithm, we first generate a set of similar documents D for a given document to provide more knowledge and ultimately improve single document key extraction. The idea behind creating a similar document set is to allow the model to use global information in addition to the local information present in any given document. To find K-nearest neighbors, they use TF\*IDF-based cosine similarity.

Following this step, they use a graph-based ranking algorithm to compute the global saliency score for each word in the word graph built on this expanded set. Because not all words in the documents are good indicators of keywords, certain syntactic filters are used during word graph construction. The edge weight between two words is calculated by multiplying the co-occurrence count of the two words across the entire document set by the similarity of the original document to the nearby concerned document, as shown in the equation:

Because this graph is based on the entire document set, it is known as the Global Affinity Graph. Once the ranking algorithm has reached a point of convergence, candidate keywords are merged to form a multi-word phrase. They use an additional rule to prune Adjective ending phrases and only select Noun ending phrases. A phrase's overall score is calculated by adding the saliency scores of individual words.

## 3. Position Rank:

The PageRank method for integrating the information about all the places of the word’s occurrence in a big text. The fundamental notion of PositionRank is to provide bigger weights (or probability) to words that are located early in a text compared to ones that appear in a later section of the document. Their algorithms primarily comprise three fundamental stages.

Graph creation at word level — They employ Nouns and Adjectives as candidates for creating nodes in their undirected word graph. Where edges connecting the nodes are based on a co-occurrence sliding window of a given size

Designing the Position-Biased PageRank — They weigh each proposed word with its opposite location in the text. For example — if a word is located in the following positions: 2nd, 5th, and 10th, then the weight associated with this word is 1/2 + 1/5 + 1/10 = 4/5 = 0.8. A vector is formed and set to the normalized weights for each potential word as shown below —

As of last, they apply the derived scores to Page Rank as mentioned above.

Formation of candidate phrases — Candidate words that have contiguous places in a text is concatenated to generate candidate phrases. They use another regex filter [(adjective)\*(noun)+], of up to length three (i.e., unigrams, bigrams, and trigrams), on top of these candidate phrases to come up with the final list of keyphrases. Finally, the phrases are rated by adding up the scores of the words that compose the phrase. (Tip — You can also play with with “New Multi-word Keyword Scoring Strategy” stated above)

## 4. Yake:

A keyword extraction approach that employs statistical characteristics to discover and rank the most essential terms. It needs just a stopwords list for it to be language neutral. The complete algorithm has 4 stages to it —

\* Preprocessing and Candidate term creation

They first do sentence-level split using segtok which is a rule-based sentence segmenter. Sentences are then separated into terms based on white space and other special characters (line break, comma, period) as the delimiters, and depending on the maximum length of keywords that we are interested in we may have 2, 3, 4 words split appropriately.

\* Featurizing Terms

Here they lay down 5 qualities to rate every unit notably Casing(Tcase: More attention to capitalized and acronyms) (Tcase: More importance to capitalized and acronyms), Position of Word in the text (Tposition: More emphasis is given to the terms that are present at the beginning of the document), Word frequency (Tnorm), Term Relatedness to Context(Trel: Checks for the variety of context in which this word appears. The better the variety, the higher the possibilities for it becoming a popular term. It may be considered as a metric to prune regularly occurring terms like stop-words) and finally Term Various Sentence (Tsentence: This characteristic assesses how often a candidate word appears with different phrases. Higher score is given to terms that regularly appear in various phrases).

\* Term Scoring

It utilizes the following algorithm to then compute the score for every unit.

\* Deduplication

It is extremely feasible to acquire comparable morphological terms when graded according to the prior scoring method. To prevent redundancy they offer a Levenshtein distance-based deduplication strategy where the objective is not to pick a word if it has a short Levenshtein distance with previously chosen terms.

\* Final Ranks

Minimum the scores, better the keywords. Pick top-k.

## 5. KeyBERT:

KeyBERT is a basic and easy-to-use keyword extraction approach that employs BERT embeddings to construct keywords and key phrases that are

most comparable to a text. Firstly, text embeddings are extracted using a pre-trained domain-specific BERT model. Secondly, word embeddings are then retrieved for N-gram words/phrases.

Finally, it employs cosine similarity as the similarity metric to discover the words/phrases that are most similar to the original material.

Top-k-rated words may then be regarded as the final collection of keyphrases.

Also, depending on the use case you can require a list of varied keywords for which you can utilize Maximal Marginal Relevance (MMR) or Max Sum Similarity.