Key Point Summarization – Parameters Summary

Supported domain_params (used when creating a new domain):

Parameter	Allowed Values	Default	Explanation
split_comments	Boolean	True	When set to False, the comments uploaded to the domain will not be split into sentences.
clean_comments	Boolean	True	By default, the service runs a minor cleansing over the comments (adding dots before newlines, removing duplicate spaces and leading dashes, cleaning common non-ascii characters and filtering comments with remaining non-ascii characters). When this parameter is set to False, the comments uploaded to the domain will not be cleaned.

Supported run_params (used when starting a summarization job):

Parameter	Allowed Values	Default	Explanation
sentence_min_len	Integer	2	Filter shorter sentences (by number of tokens)
sentence_max_len	Integer	50	Filter longer sentences (by number of tokens)
mapping_policy	“STRICT”, ”NORMAL”, “LOOSE"	“NORMAL”	Policy for determining if a sentence is matched to a key point: for “strict”, only pairs with very high matching scores will be considered matched, leading to a higher precision and a lower coverage, and vice versa for loose.
kp_granularity	“FINE”, “NORMAL”, “COARSE”	“NORMAL”	Used for key point selection: choose “FINE” for more fine grained key points, and “COARSE” for more distinct key points.
min_matches_per_kp	Integer	5	Minimal number of matching sentences per key point (key points with less matches will be discarded)
max_num_kps	Integer	200	Upper bound on the number of key points to return.
invalid_kps_comment_ids	List of strings:	Empty list	A list of comment_ids whose sentences should not be selected as key point candidates
kp_min_len	Integer	3	Minimal length of key points (by number of tokens)
kp_max_len	Integer	8	Maximal length of key points (by number of tokens)
key_points **	List of strings	None	Let the service match the sentences to the given key points instead of extracting them automatically. This enables a human-in-the-loop scenario, where the automatically extracted key points are reviewed by the user, and the sentences are then remapped to the revised keypoints.
key_points_by_job_ids **	List of strings	None	It is also possible to use key points from previous jobs by supplying the job_ids in this param.
key_point_candidates **	List of strings	None	When key_point_candidates are provided, the service matches the sentences to the provided key_point_candidates but also automatically extracts new ones, if exist.

key_point_candidates _by_job_ids **	List of strings	None	It is also possible to use key point candidates from previous jobs by suppling the job_ids in this param. The final key points from these jobs will serve as key_point_candidates.

** Only one of these parameters should be use