SimSage Search Syntax and AI

Introduction

This document explains the advanced query syntax used by SimSage to find items across documents and their metadata. SimSage has a sophisticated semantic search engine that automatically uses relationships, probabilities and distances between concepts and words to find the best results. SimSage can also perform searches within searches, and search for complex entities like credit-card numbers, social security numbers, names of people, countries, and many other entities.

SimSage searches in the body of documents by default, using concepts or words found in document titles (where possible) to emphasize the search score / importance of a document.

Beyond SimSage Search there is SimSage AI. A powerful integration of SimSage with Google or OpenAI Large Language Models. This will save you a lot of time and effort digesting your information.

Search operators

Double Quote Operator

This operator indicates an exact search, i.e. words without relationships, and close together.

Example

> “market value“

Title Searches

Documents usually have titles. SimSage will set the title of a document to its filename (without any path) if it cannot find a title. Where a document has a title, SimSage can use the “intitle:” directive to look for matching words or concepts.

Example

> intitle: baseball game

Entity Searches

Entities are known objects in SimSage such as people’s names, credit-card numbers, and many more (see table below). Entity searches enables searching for more general terms. add `entity: <entity-name>` to your query. Where `entity-name` can be one of:

entity name	description
nin	UK national insurance number
ssn	US social security number
credit-card	A valid credit-card (Lunn verified) number
ip-address	an IPv4 or IPv6 address
mac-address	a network Media Access Control address
url	An HTTP or HTTPS based web address
person	A person (e.g. Jimmy Smith)
email	an email address
city	biggest cities in the world
country	all countries of the world
company	a small set of known companies
brand	a small set of known brands
continent	all continents of the world
capital	all capitals of the world
state	all states of the US
phone	valid phone numbers for the US and UK

zip	US zip codes
postcode	UK post codes
hashtag	hashtags, (e.g. #test, #market)
secret	A large set of secrets as defined by the truffleHog regex set (dxa4481/truffleHogRegexes)
hip	SimSage home insurance policy numbers
pip	SimSage personal insurance policy numbers
cip	SimSage car insurance policy numbers
policy	SimSage policy numbers

You must use the exact “entity name” when using an “entity:” search. Your search will fail if you do not pass a known entity name after the “entity:” keyword.

Example

> entity: credit-card
>
> entity: mac

URL Searches

URLs in SimSage are the primary keys of whatever data type you’re searching for. In some cases these aren’t actually URLs. Websites, and most web-based systems do use URLs. The *inurl:* <many words> can be used to look for a series of words occurring inside the URLs / primary keys of your data.

Example

> inurl: research jobs
>
> inurl: research facilities in Japan

In text / main content of your data searching

This filter is provided to enable a user to switch back to the default search of searching inside document body text. This is SimSage’s default for searching and does not need to be specified ordinarily. However, it can be explicitly used in combination with other filters to switch between them. The syntax for this search is: *intext:* <many words>

Example

> intext: the effects of radiation

Metadata searches

Arbitrary name values can be searched too. If your data provides, for instance, a “status” field in its metadata you can search for a metadata-name equals value. How metadata is mapped, and what metadata is available and what it is set too all depend on your sources.

In our example we assume there is a metadata field called “status” and it has values like “closed”, “released”, “open” etc.

Example

> status: closed

Source Filters

In many cases your SimSage system can have many “sources”. A source is where your information comes from / external integration points. These sources will have been given names by your administrator and can be referenced as part of a filter using the source: keyword. The usage is *source:* <unique name of a source>. Your search will be rejected with an error if the source name is incorrect or does not exist.

Example

> source: second floor server
>
> source: google drive one

Exclude Filters

You can chose to exclude a single word / concept by prefixing the word with a hyphen (-) as shown in the example below.

Example

> -second

This operator applies to inurl / allinurl / intitle / allintitle items too. The operator only applies to the exact word, not its relationships if applicable.

Time Based Searching

We will group these into one category. Time based searches are modeled after the Google time based searches and must have this exact syntax:

- before: yyyy-mm-dd
- after: yyyy-mm-dd

Example

> before: 2020-05-02

Searches within Searches

You can search again within the results of a search. SimSage uses the “sub” keyword to add searches within searches. The second search isn’t strictly tied to the search that came before it but will try and find the closest result to the previous search. A document is rejected if it does not match all requirements of each and every sub-search. A sub search is an “infix” operator, meaning that it appears between two search directives as defined by the rest of this document.

Syntax:
some search terms **sub** *some other search terms*
some search terms **contains** *some other search terms*

some search terms **and** *some other search terms*
some search terms **with** *some other search terms*

You can create up to nine sub-searches in a row. However, each search counts as a new additional search and requires additional processing power of the SimSage platform (equivalent to another search each time).

Example

- Suppose you wanted to find an email address for "John Smith". You can first search for "John Smith" and then instruct SimSage to find the closest email address to any of those terms found by using the "sub" keyword and the "entity: email" selector. Search for:
  - *John Smith sub entity: email*
    - SimSage will return no results if it cannot find any email addresses or John Smith in any combination together
    - SimSage will return the closest email address it can find to John Smith. This might not be John Smith's email address depending on the data.
- Suppose you want to see if there are any sensitive MAC addresses in your router documentation. You know all your router documentation has the word "router" in the title of each document, so you can then search for
  - *intitle: router with entity: mac*
- Suppose you want to find documents where people, email addresses AND social security numbers are mentioned. You can do a triple "sub" search by entering:
  - *entity: person with entity: email and entity: ssn*
    - SimSage effectively executes three searches and needs to do 3x the amount of work
    - The "sub, and, contains, with" keywords act as an AND across the content of the document. Any document not containing all three will not be shown.
    - SimSage will show only the first "person" it finds in each document followed by the closest email and social security number relative to that person.

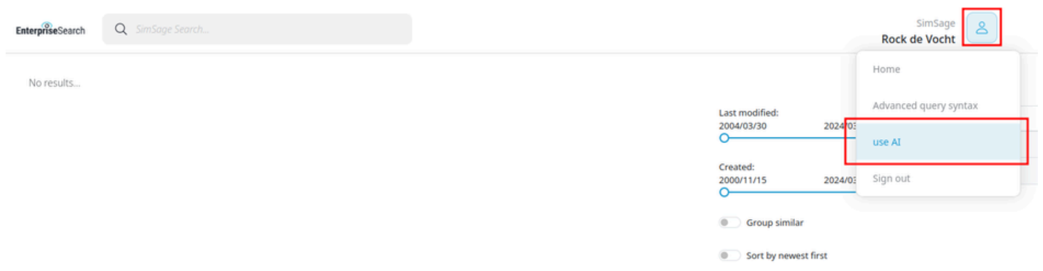
> John Smith sub entity: email
>
> intitle: router and entity: mac
>
> entity: person and entity: email with entity: ssn

SimSage AI

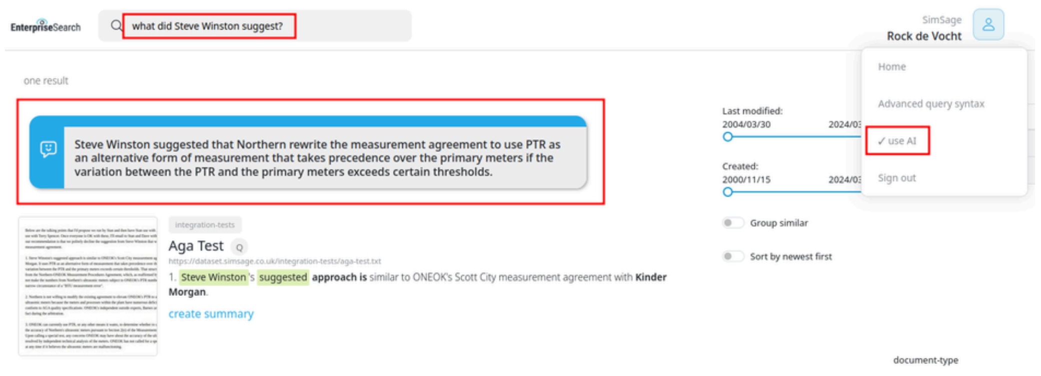
SimSage can use its search engine result output with powerful language models such as Google's Gemini and OpenAI*. This process uses Retrieval Augmented Generation (RAG) and passes snippets of your search results to Large Language Models (LLMs) along with your search query.

In addition SimSage can use these LLMs to go into a Question and Answer mode for any of your documents where you can interrogate the content of a document found in your searches.

NB. the "use AI" menu will only show if AI has been enabled on your platform.



Once the AI is enabled (which can be either Google Gemini or OpenAI depending on your platform configuration) SimSage will use their LLMs to pass small snippets of text of your search results and try and answer the user's query.

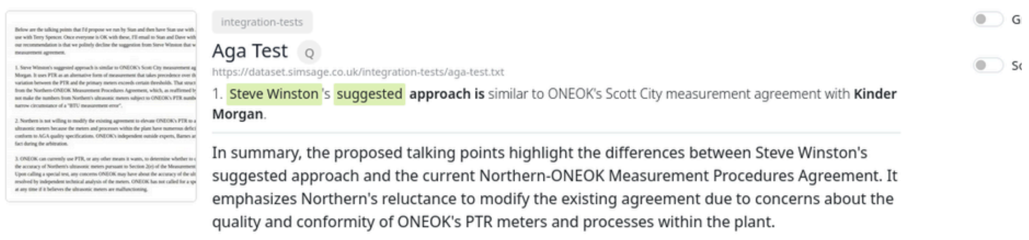


Steve Winston suggested that Northern rewrite the measurement agreement to use PTR as an alternative form of measurement that takes precedence over the primary meters if the variation between the PTR and the primary meters exceeds certain thresholds.

Aga Test Q
https://dataset.simsage.co.uk/integration-tests/aga-test.txt
1. Steve Winston's suggested **approach is** similar to ONEOK's Scott City measurement agreement with **Kinder Morgan**.

Note the "create summary" link at the bottom. Clicking that link will create a summary of the document's content. How much content it passes to Google Gemini or OpenAI depends on

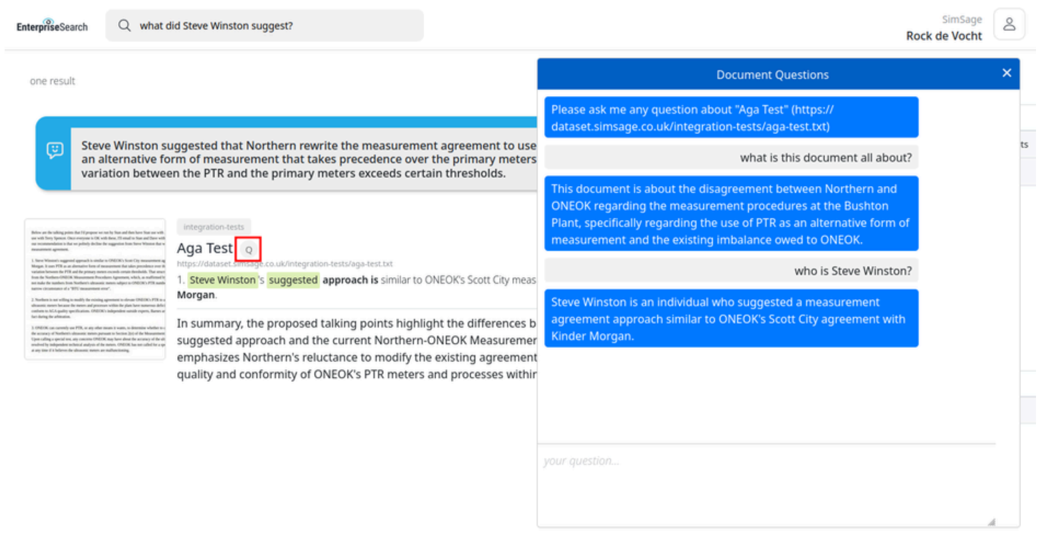
- how the SimSage platform is configured
- the limits of the language model used



Aga Test Q
https://dataset.simsage.co.uk/integration-tests/aga-test.txt
1. Steve Winston's suggested **approach is** similar to ONEOK's Scott City measurement agreement with **Kinder Morgan**.

In summary, the proposed talking points highlight the differences between Steve Winston's suggested approach and the current Northern-ONEOK Measurement Procedures Agreement. It emphasizes Northern's reluctance to modify the existing agreement due to concerns about the quality and conformity of ONEOK's PTR meters and processes within the plant.

The "Q" icon shown to the right of the title can be clicked to go into document Question and Answer mode.



Please ask me any question about "Aga Test" (https://dataset.simsage.co.uk/integration-tests/aga-test.txt)

what is this document all about?

This document is about the disagreement between Northern and ONEOK regarding the measurement procedures at the Bushton Plant, specifically regarding the use of PTR as an alternative form of measurement and the existing imbalance owed to ONEOK.

who is Steve Winston?

Steve Winston is an individual who suggested a measurement agreement approach similar to ONEOK's Scott City agreement with Kinder Morgan.

[*] *if enabled on your SimSage platform*