Proximity searching in databases

# Introduction

Proximity searching allows you to look for keywords that appear close together in a piece of text. Some databases will even allow you to set a specified number for the range that words appear from each other.

The number of results retrieved using proximity search are much less and more relevant than a simple ‘**AND**’ search. However, proximity searches are less precise than a phrase search. How you conduct a proximity search will vary depending on the database you’re searching and there are often two types of proximity search available.

* **Type 1**: allows you to specify the order the words appear in.
* **Type 2**: only allows you to specify the proximity.

Because proximity searches are quite specific, expect to retrieve fewer results.

**💡 Tip:** If you’re unfamiliar with search operators, try looking at online guide to using [search operators in your search](https://www.education.library.manchester.ac.uk/mle/search-operators/story_html5.html).

# Making use of proximity search

If you search for graphic novels, your search could find articles where the words graphic novels appeared next to eachother, or anywhere in the article, for example, graphic in one sentence/paragraph and novels in another.

However, a proximity search allows you to specify how near one term should appear from another. For example, in a database where ADJ is used for proximity this example search

‘graphic ADJ3 novels’

…shows that the first word should be within three words of the second word in any order.

You can also use Proximity search in combination with other search operators such as Boolean, as shown in the example below:

‘graphic ADJ3 noivels AND science fiction OR sci-fi’

# How to proximity search in popular databases

## OVID (e.g. MEDLINE, EMBASE, PsycINFO)

The proximity command in OVID is ‘**ADJn**’ where ‘**n**’ represents the number of words in between your keywords (minus one, so ADJ3 will have two or fewer words in between). The retrieval is irrespective of the order in which you have entered the terms. Please note that a free text phrase search (using “quotation marks”) will search for your words in the exact order inputted.

Example:

Searching OVID for; ‘**Stroke patients** ADJ3 **depression**’

Retrieves:

* **Stroke patients** with **depression**
* **Depression** in **stroke patients**
* **Depression,** and functional outcomes in **stroke patients** – although at first glance this looks more than three words the database will ignore certain stop words such as ‘**and**’ and ‘**in**’. Consult the help guide in the database to find a full list of stop words.

## EBSCO (e.g. CINAHL, British Education Index, Anthropology Plus, etc.)

There are two different proximity commands in EBSCO.

1. The **NEAR** command **Nn** (where the lowercase **n** represents the number of words that could appear in between your keywords.) The retrieval is irrespective of the order you have entered the terms.

Example:

Searching EBSCO for; ‘**Treatment** N2 **Screening’**

Retrieves:

‘…**screening**, and **treatment** of depressive…’

and

‘Harms of **Treatments**, **Screening**, and Tests’

1. There is also the **Within** command **Wn** (where the lowercase **n** represents the number of words that could appear in between your keywords.) It only retrieves articles/bibliographic records in which it finds the keywords in the order in which they are input.

Example:

Searching EBSCO for; ‘**Treatment** W2 **Screening’**

Retrieves:

‘**screening**, and **treatment** of depressive…’

But it *wouldn’t* retrieve… ‘Harms of **Screening**, **Treatments**, and Tests’

## Web of Science

Web of Science uses the Near command (**NEAR/n**) for proximity searches. The NEAR command does not allow you to set the order of the keywords being searched.

Example:

Searching Web of Science for; ‘**Empower** NEAR/2 **individual**’

Retrieves:

* ‘Conceptualising **empowerment** in **individuals**’
* **‘Individual** rights by **empowering**’

## Scopus

Scopus also has two different types of proximity search available.

1. The first type of proximity search in Scopus is the **Within** command. Types as **W/n**, where the lowercase **n** represents the number of words that could appear in between your keywords. It does not allow you to specify the order that the words appear in.

Example:

A within search in Scopus for ‘**Breast** W/2 **cancer**’

Retrieves:

* ‘**Breast cancer** in men and women’
* ‘**Cancer** of the **breast**’

1. The second type of proximity search in Scopus is ‘**Pre**’. Use the **Pre/n** command where the first word must be no more than (**n**) words apart from the second word.

Example:

A search in Scopus for; ‘**Breast** Pre/2 **Cancer**’

Retrieves:

* ‘**Breast Cancer**’ but not the ‘**Cancer** of the **breast**’

## Proquest (e.g. ASSIA)

There are three different commands for proximity searching in Proquest databases. These are ‘**NEAR/**’ and ‘**N/**’ and ‘**PRE/**’.

1. You use ‘**NEAR/**’ and set a number range (word order not set.)

For example, ‘**women** NEAR/3 **education**’.

1. You can also use ‘**N/**’.

For example, ‘**women** N/3 **education’.**

Please note that if you use **NEAR** without specifying a number, the default number will be applied, i.e., within four words of each other.

The letter **N** without a number will not work, the N will be treated as a search term and not a command.

Example:

A search in Proquest for; ‘**women** NEAR/3 **education**’ and ‘**women** N/3 **education**’

Retrieves:

* ‘**women** continue undertaking **education…**’ and
* ‘**women** that think **education…**’

1. **PRE/**(including a number) allows you to search for a word that proceeds another within a specific range.

Example:

A search in Proquest for; ‘**cancer** PRE/4 **screening**’

Retrieves:

* ‘**cancer** avoidance through **screening**’

## Cochrane Library (Wiley version)

Cochrane Library uses the near command to carry out proximity searches. This looks like ‘**NEAR/n**’ where **n** is a number. Cochrane does not allow you to set the word order.

Example:

A search in Cochrane for; ‘**Lung** NEAR/4 **cancer**’

Retrieves:

* ‘**Lung cancer**’
* ‘**Cancer** of the **lung**’

You can set word order by using NEXT but this assumes the words are next to each other.